Cryptographic Key Management for the Vehicles of Tomorrow

by

Christopher Richard Allden Hicks

A thesis submitted to
The University of Birmingham
for the degree of
Doctor of Philosophy

School of Computer Science
College of Engineering & Physical Sciences
The University of Birmingham
July 2019

Abstract

The automotive industry is undergoing a major transformation process in which nearly every part of the vehicle is becoming digital and connected. Modern vehicles are often connected to the internet, feature several wireless interfaces and will soon communicate directly with surrounding vehicles and roadside infrastructure using Vehicle-to-Everything (V2X) technology. However, this transformation has not yet been paralleled by the development of techniques or standards which address the cyber security challenges posed by these systems.

In this thesis, we present several new cryptographic and key management flaws in an existing automotive immobiliser system and we develop two new V2X architectures for improving the safety and privacy of tomorrow's connected and autonomous vehicles. Specifically, we study the AUT64 automotive block cipher and its associated authentication protocol in a real-world immobiliser system. Despite having a 120 bit key, we find a number of flaws in the system which we combine to present several practical key-recovery attacks.

Our first new V2X architecture, IFAL, provides a practical and secure improvement to the leading European standard for Vehicle-to-Everything (V2X). IFAL introduces a new certificate issuance mechanism that eliminates the trade-off between pseudonym duration and bandwidth. Our second architecture, VDAA, addresses the need for efficient techniques that preserve vehicle privacy despite dishonest or colluding certificate authorities.

Contents

Chapter 1

Introduction

Vehicles are an important and pervasive part of everyday life. In the UK alone there were 3,169,760 new vehicle registrations in 2017 [232], contributing to a national total of almost 40 million [231]. During the same period the automotive industry added over £20 billion in value to the UK economy and provided more than 855,000 jobs [219]. Vehicles not only strengthen economies and provide jobs, they also play a decisive role in shaping the development of society by defining the speed with which goods, services and people can move from one place to another. Private vehicle ownership provides a degree of personal control and autonomy which cannot be found elsewhere [79] and, for older drivers in particular, driving is often the only option for independent mobility [194] and is considerably safer than walking or cycling [182].

It is critical that vehicles are safe and that they are protected from theft. Despite Britain having some of the safest roads in the world [108], vehicle related accidents are consistently a leading cause of death amongst the 5-19 and 20-34 age groups [122]. Globally, 1.35 million people are killed and 50 million are injured on the world's roads every year [108]. Technologies which make vehicles safer and the standards and legislation which support this are therefore crucial developments which address a critical and global public health challenge, a barrier to human development and economic growth.

Vehicles are expensive and often left unattended. Over the last decade, vehicle-related theft has accounted for around one in seven reported crimes in England and Wales and around 1 in every 25 vehicle owning households has been a victim [95]. Recent crime data for England and Wales indicates a significant rise in the number of vehicle thefts (e.g. 106,210 in 2017 compared to 76,163 in 2015) [230] whilst also showing a decline in the proportion of thefts resulting from forced locks (e.g. 25% in 2015 compared to 7% in 2017) [66]. This data is thought to reflect a rise in the use of increasingly advanced techniques aimed at disabling or abusing the electronic systems which protect the vehicle [105].

The automotive industry is undergoing a major transformation process in which nearly every part of the vehicle is becoming digital. Modern vehicles are controlled, made safe and kept secure by a large network of microcontrollers and Electronic Control Units (ECUs) that are fully programmable [103]. Modern vehicles are often connected to the internet and feature wireless interfaces for connecting with consumer devices. The digitalisation of modern vehicles has enabled the development of semi-autonomous safety features like Advanced Emergency Braking (AEB) and Lane Departure Warning (LDW) systems which use radar and video sensors to prevent accidents by opportunistically taking control of the vehicle to prevent or minimise any potential collision. It is estimated that AEB could save more than 1,000 lives every year within the European Union (EU) alone [135] and it is proposed to be mandatory on all new vehicles sold within the EU from 2022 [97]. Electronic vehicle immobilisers are estimated to have reduced the rate of vehicle theft by 40% [243] and have been mandatory in all new passenger cars sold within the EU since 1998 [55].

The vehicles of tomorrow will have even greater connectivity and will have advanced autonomous features which enable them to operate with little to no human input [169]. Tomorrow's connected vehicle will communicate directly with surrounding vehicles and roadside infrastructure to provide new efficiency and safety features such as vehicle

platooning, collaborative forward collision warning and emergency electronic brake lights [246]. In the near term, highly connected vehicles are perceived as a key enabler for autonomous driving as they allow for new semi-autonomous safety features based on additional situational awareness. Tomorrow's fully autonomous self-driving vehicles have the potential to significantly improve upon the safety and efficiency of transportation, both saving lives and helping to preserve the environment [1]. Fully autonomous vehicles will address these two critical global challenges by eliminating the cause of 94% of all vehicle crashes, driver inattention [10], and by providing up to 87% fuel savings based on the shared ownership of electric vehicles [32]. Autonomous vehicles are additionally expected to enhance mobility for the young, elderly and disabled, and to provide an associated improvement in access to education, employment and healthcare in these groups [56].

As vehicles become more digital, programmable and connected they also become vulnerable to a new type of adversary. Despite Tesla's autopilot feature already being used on public roads in the UK [54] and Lyft already offering rides in self-driving vehicles at select U.S. locations [165], the rapid transformation of modern vehicles has not yet been paralleled by the development of techniques or standards which adequately address the cyber security challenges posed by these systems. The automotive industry has historically failed to use secure cryptography or appropriate key management techniques [126, 105, 249, 248, 28] and there is no sign that things have improved. The current state of the art in automotive security has even been compared to computers in the early days of the Internet [103]. There is a clear requirement for new research which identifies the cyber security flaws in existing automotive systems and which proposes new techniques for securing the vehicles of tomorrow.

1.1 Contributions

This thesis advances the state of the art in electronic automotive security and safety systems. Specifically, the contributions are as follows

1.1.1 Dismantling the AUT64 Automotive Cipher

AUT64 is a proprietary 64-bit block cipher with a 120-bit secret key used in a number of automotive security applications which include vehicle immobilisation and remote keyless entry systems. We present full details of AUT64 including a complete specification and analysis of the block cipher, the associated authentication protocol, and its implementation in a widely used vehicle immobiliser system which we have reverse engineered. The AUT64 block cipher is of special cryptographic interest because it has an unusual, unbalanced Feistel network design [211] and also because it offers security which is dependent not only on the secrecy of the private key but also on its value.

We identify a number of unique cryptographic weaknesses in the design of the AUT64 block cipher and furthermore reveal a significant weakness in the key diversification methods used by the immobiliser implementation. We present two key-recovery attacks based on the cryptographic weaknesses which, when combined with the implementational key diversification weaknesses, break both 8 and 24 round configurations of AUT64. Our attack on 8 rounds requires 512 plaintext-ciphertext pairs and, in the worst case, just $2^{37.3}$ offline encryptions. In most cases our attack can be executed within milliseconds on a standard laptop. Our attack on 24 round AUT64 requires only 2 plaintext-ciphertext pairs and $2^{48.3}$ or fewer encryptions to recover the 120-bit secret key.

Dismantling the AUT64 automotive cipher is an important contribution to the body of research which identifies flaws in existing vehicle immobiliser and remote keyless entry solutions [105, 249, 248, 28]. We identify the steps which can be taken to mitigate the present threat and provide additional motivation for the automotive industry to move

towards standardised algorithms and peer-reviewed protocols.

1.1.2 IFAL: Issue First Activate Later Certificates for V2X

In the very near future, vehicles will directly communicate with each other and with roadside infrastructure. V2X communication is expected to substantially improve upon road safety and traffic efficiency by allowing road users to make optimised decisions on the basis of an enhanced and collaboratively formed awareness of the surrounding environment [1]. To enable the coordinated and internationally interoperable deployment of V2X there is a need for common standards, the adoption of which is even a legal requirement for EU member states [72].

In this thesis we present Issue First Activate Later (IFAL) certificates, a practical and secure improvement to the leading European standard for V2X communication. The leading standards for V2X all use time-limited pseudonym certificates as a means of protecting the long-term privacy of road users. Vehicles periodically change their pseudonym certificate in the hope that their long-term behavioural patterns are not revealed. Pseudonyms for V2X are technically challenging because they multiply the number of identities belonging to each vehicle and correspondingly increase the latency of revocation.

IFAL incorporates a novel cryptographic key management technique that both avoids the need for certificate revocation and which provides additional support for vehicles with limited and intermittent connectivity. Our new construction is equivalent to symmetric key-diversification in the public key setting and allows for the time-delayed activation of pre-issued vehicle pseudonym certificates. IFAL improves upon the current approach of using time-limited certificates by eliminating the need to compromise between the bandwidth required for transferring certificates and the privacy afforded by the time-limit of each certificate. In practice, IFAL may save up to 288 KB of bandwidth per day and is the difference between a vehicle requiring a cellular data subscription or not.

We introduce a new formalisation of the V2X security and privacy requirements from the standard which we apply to IFAL to show that it is a provably secure and privacy conscious V2X scheme. In addition, we demonstrate that IFAL is practical with an analysis of the computational complexity and by presenting the results of our reference implementation which shows that a 5 year pseudonym certificate file can be computed, on average, in 9.03 seconds using a standard desktop computer.

IFAL is a new certificate issuance technique for ensuring the safety and privacy of tomorrow's connected and autonomous vehicles. IFAL is already mentioned in a recent pre-standardisation study on pseudonym change management [84] by the leading European standards body. If adopted into the main standards, IFAL could contribute to the secure and privacy conscious deployment of V2X throughout Europe and allow for the corresponding improvements in road safety and efficiency to be realised.

1.1.3 VDAA: A Vehicular DAA Scheme for ECDSA Pseudonyms in V2X

The leading V2X architecture standards [85, 29] both use the role-separation of certificate authorities as a mechanism for vehicle privacy during certificate issuance. In essence, both standards propose an enrolment authority which issues long-term certificates and a pseudonym authority which issues short-lived pseudonym certificates. Vehicles authenticate to the pseudonym authority using a long-term enrolment credential which does not reveal their real identity. Whilst both standards include procedural privacy measures which provide some protection against insiders who attempt to link pseudonym certificate values with real vehicle identities, neither architecture adequately protects against compromised, dishonest or colluding certificate authorities. The European Data Protection Working Party have specifically called for new techniques which limit the risk of collusions between certificate authorities in V2X [183].

In this thesis we present a new architecture which cryptographically addresses the risk of colluding certificate authorities. We introduce our new Vehicular DAA (VDAA) scheme which reconciles the strong privacy guarantees of Direct Anonymous Attestation (DAA) with the efficiency and low-latency that is required for V2X broadcast messaging. Early V2X field studies [214] identified that vehicles must be able to verify at least 1,000 signatures per second in order to deal with busy intersections. The ECDSA signature scheme was selected as the algorithm most able to meet this requirement given the anticipated computational constraints of first and second generation connected vehicles. ECDSA signatures on V2X broadcast messages has since been adopted by the leading international standards. Unlike many other schemes which apply DAA to V2X, VDAA uses regular ECDSA signatures on broadcast messages which means that verification latency is unaffected.

Our VDAA scheme implements DAA credentials such that vehicles are able to make anonymous and unlinkable requests for pseudonym certificates whilst still allowing for centralised revocation based on malicious V2X broadcast messages. Vehicles are prevented from abusing their anonymity with a novel DAA attribute construction that restricts each vehicle to requesting one pseudonym per certificate time period. Vehicles which attempt to request multiple pseudonyms for the same period are discovered, forfeit their unlinkability and may also be prevented from making further requests. We present a new security model for VDAA and show that the unforgeability and unlinkability of our ECDSA broadcast messages can be reduced to the security of the underlying DAA scheme.

VDAA uniquely addresses the need for new techniques which limit the risk of colluding V2X certificate authorities [183] whilst also retaining the architecture, ECDSA broadcast message signatures and centralised authority over vehicle revocation which are necessary for standards compliance.

1.2 Overview

In this thesis we present several new cryptographic and key management flaws in an existing automotive immobiliser system and introduce two new schemes for improving the security and privacy of tomorrow's connected and autonomous vehicles. We structure the thesis as follows

Chapter 2 Introduces the relevant technical background on the design and analysis of digital automotive security and safety systems. We also provide the terms and definitions which allow for a complete understanding of the later chapters.

Chapter 3 Presents the literature and standards which fully contextualise the contributions in this thesis. In particular, we expound upon the challenges in cryptographically secure automotive system design and detail the state of the art in connected vehicle public key infrastructure. We identify and relate the relevant work from the broader literature on cryptographic key management and digital privacy.

Chapter 4 Provides a thorough analysis of a popular and previously unstudied vehicle immobiliser system. We identify a number of cryptographic flaws in the proprietary block cipher design and an additional flaw in the key management method. We show that the design is unsuitable for continued deployment and motivate the transition to modern, standardised algorithms and peer-reviewed protocols. This chapter is based on the following publication

Christopher Hicks, Flavio Garcia, and David Oswald. Dismantling the AUT64 Automotive Cipher. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(2):46–69, May 2018.

Chapter 5 Introduces Issue First Activate Later (IFAL) certificates, a new certificate issuance method that improves upon the leading European standard for V2X architecture.

IFAL incorporates a novel public key construction which allows for the time-delayed activation of pre-issued vehicle pseudonym certificates and reduces the trade off between privacy and bandwidth from the standard. We show that our IFAL scheme is practical, standards compliant and, in a formal setting, both secure and privacy conscious. This chapter is based on the following publication

Eric Verheul, Christopher Hicks, and Flavio D. Garcia. IFAL: Issue First Activate Later Certificates for V2X. In *IEEE European Symposium on Security and Privacy, EuroS&P*, June 2019

Chapter 6 Presents our new Vehicular DAA (VDAA) scheme for V2X which improves upon the leading standards by both preventing certificate authority collusion and maintaining vehicle privacy despite compromised trusted hardware. VDAA reconciles the strong privacy guarantees of DAA with the efficiency and low-latency of ECDSA broadcast messages signatures. We evaluate VDAA in a formal setting and show that the unlinkability and unforgeability of our broadcast messages can be reduced to that of the underlying DAA scheme. Additionally, VDAA introduces a novel DAA attribute construction which prevents vehicles from abusing their anonymity to request multiple pseudonyms for the same certificate time period. This chapter is based on the following publication

Christopher Hicks and Flavio D. Garcia. VDAA: A Vehicular DAA scheme for ECDSA Pseudonyms in V2X. In *IEEE European Symposium on Security and Privacy, EuroS&P*, September 2020

Chapter 7 Concludes this thesis by reflecting on the contribution and identifying the key areas for future work.

Chapter 2

Background & Preliminaries

This thesis presents new research which identifies the cyber security flaws in existing automotive systems and proposes novel techniques for securing the vehicles of tomorrow. In this chapter, we set the stage for our research by introducing the technical background, terms and methods of analysis which contextualise the cryptographically secure design and evaluation of electronic automotive systems.

2.1 Vehicle Theft Prevention Systems

The first mainstream automotive theft-prevention technique was steering locks which were introduced in the 1960s to tackle the emerging problem of vehicle theft. Whilst steering locks did initially stymie the problem, motivating the development of central locking and car alarm systems, it was not until the deployment of electronic vehicle immobilisers in the early 1990s that crime levels first began to decline [177].

2.1.1 Electronic Vehicle Immobilisers

Electronic vehicle immobilisers have more evidence for their effectiveness than any other vehicle theft prevention mechanism. In the UK alone, it is estimated that electronic immobilisers prevented 4 million vehicle thefts between 1992 and 2013, representing 43% of the decline in thefts witnessed during the same period [177]. In recognition of their

effectiveness, electronic immobilisers have been mandatory for all new vehicles sold within the EU since 1998 [55]. It is estimated that more than 98% of all vehicles registered in Great Britain are fitted with one [177].

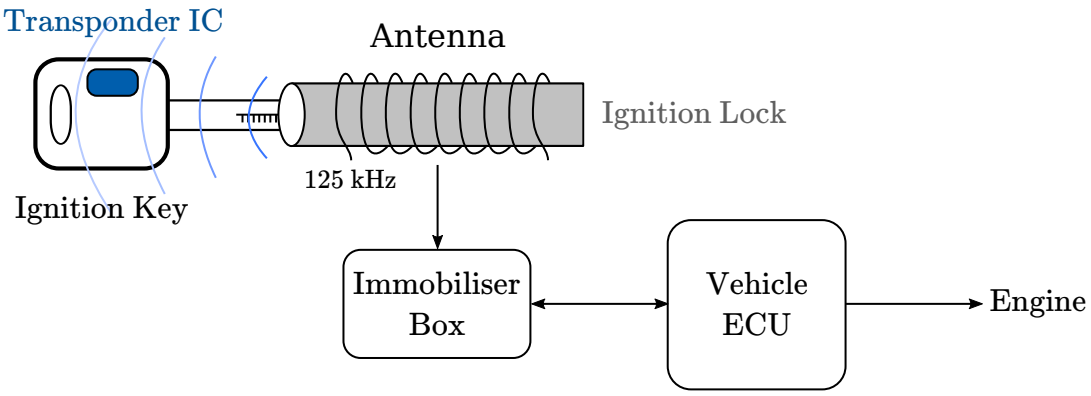


Figure 2.1: A typical electronic vehicle immobiliser system.

A vehicle immobiliser is any device which prevents the unauthorised use of a vehicle by ensuring that the engine may not be started and that the vehicle may not be driven or moved under its own power unless the device is disabled. A typical electronic immobiliser, shown in Figure 2.1, comprises a Radio Frequency IDentification (RFID) transponder chip embedded in the vehicle ignition key, an antenna coil placed around the ignition barrel and an immobiliser box which is connected to the vehicle ECU. Immobiliser transponders are coded so that the immobiliser box will only be disabled after communicating with a specific transponder. All modern immobilisers use cryptography for authentication between the transponder and the vehicle immobiliser, typically based on a symmetric key challenge-response protocol [105].

2.1.2 The Analysis of Electronic Vehicle Immobilisers

By the time the EU immobiliser legislation was published in 1995 [55], the Data Encryption Standard (DES) symmetric encryption algorithm was a long-established open standard backed by twice-reaffirmed U.S. government approval [64]. The DES algorithm was

already widely used to secure commercial transactions over the internet [7] and to encrypt satellite television audio streams [115]. DES provided a focus around which the field of cryptanalysis could gather pace, and new techniques for analysing the security of DES and other ciphers were soon realised. Differential and linear cryptanalysis techniques were developed which were able to reduce the computational complexity of breaking DES to less than that of exhaustive key search [171, 18]. By the end of the decade, it was possible for a special-purpose distributed system to break a DES key in less than a day [174] and the Advanced Encryption Standard (AES), the most widely used symmetric key encryption algorithm of modern times, was well on its way towards standardisation [62].

Despite the relatively widespread use of standardised cryptographic algorithms like DES and the maturity of cryptography as a research field, the EU immobiliser legislation only specified that electronic immobiliser systems need to have at least 50,000 variants and that they incorporate rolling codes. For comparison, a DES key which was already considered weak at the time has 72,057,594,037,927,936 possible values. It is evident from the legislation and the later efforts to reverse engineer and study vehicle immobiliser systems that they were implemented without consideration for the cryptographic standards of the time. For many years, only weak, proprietary cryptography was implemented in automotive immobiliser systems worldwide [105]. Owing to the obscurity of their implementation, many of the techniques for evaluating proprietary automotive cryptography were first developed from the cryptanalysis of popular standardised algorithms such as DES.

2.2 Evaluating Symmetric Encryption Primitives

DES, AES and the proprietary automotive algorithms which feature in this thesis are all symmetric encryption schemes, distinguished by their use of the same secret for both encryption and decryption. Encryption schemes are primarily designed as confidentiality mechanisms, they address the problem of providing secret communication over an insecure

channel, but they can also be used as building blocks for providing other services. All modern electronic vehicle immobilisers provide an authentication mechanism which is based on the security of an underlying encryption scheme.

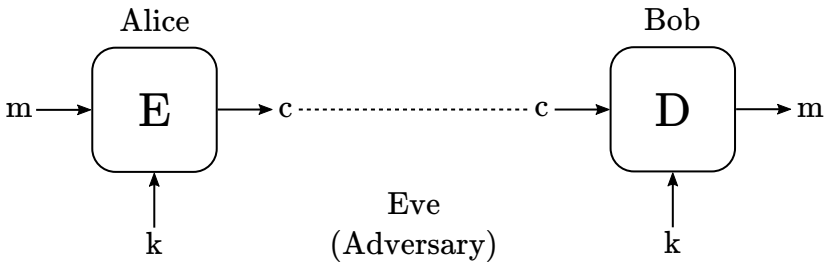


Figure 2.2: Private-key encryption setting.

Evaluating the security of an encryption scheme first requires formalising the setting in which it is expected to be secure. As shown in Figure 2.2, the basic setting consists of a sender Alice, a receiver Bob and an insecure channel that may be tapped by an adversary Eve. The goal is to allow Alice to communicate information to Bob over the insecure channel without letting the adversary Eve learn what was communicated. The secret information that Alice wishes to communicate to Bob is denoted the *plaintext* m and the public information sent over the insecure channel is denoted the *ciphertext* c. The secret information that allows Bob to recover the plaintext from the ciphertext is denoted the *key* k. Formally, an encryption scheme is defined as follows.

Definition 2.2.1 (Symmetric Encryption Scheme) A symmetric encryption scheme is a triple (G, E, D) of probabilistic polynomial-time algorithms satisfying the following two conditions:

1. On input a security parameter 1^n, the key generation algorithm G outputs a key k.
2. For every key k in the range of $G(1^\eta)$, and for every message $m \in \{0,1\}^*$, the encryption algorithm E and the decryption algorithm D satisfy the following consistency equation:

$$D\Big(k, E(k,m)\Big) = m$$

There are two approaches for evaluating the security of an encryption scheme. The classical approach is information-theoretic, based on the seminal work of Claude Shannon [216]. Shannon lay the foundations for modern cryptography with his rigorous and formal approach to defining the general mathematical structure and properties of secrecy systems. Information-theoretic security is concerned with the unconditional security of a cryptosystem. Even against an adversary with unlimited computational power, a *perfectly secure* or information-theoretically secure cipher is one for which the ciphertext reveals absolutely nothing about the plaintext. Unfortunately, one of the conditions for perfect secrecy is that the secret key used to protect the message must be at least as long as the message. Information-theoretic security is therefore of limited applicability to modern cryptosystems as they are often required to encrypt huge amounts of information, necessitating a mechanism which could securely transfer an equally huge number of key bytes.

The modern, pragmatic method for evaluating the security of an encryption scheme is based on computational complexity. The computational complexity approach is concerned with, rather than whether it is theoretically possible, whether it is computationally feasible for an adversary to learn anything about the plaintext from the ciphertext. The computational complexity approach allows for the design of symmetric encryption schemes that are secure despite short, practical key lengths and also allows for the realisation of secure public-key cryptosystems which cannot exist under the information-theoretic approach [109]. The computational complexity equivalent of Shannon's perfect secrecy is called *semantic security*. A cryptosystem is semantically secure if it is infeasible for any adversary to learn anything about the plaintext, other than its length, from the ciphertext. An equivalent definition[1], termed *ciphertext indistinguishability*, is that no adversary can distinguish between the encryptions of a given pair of plaintexts [110].

[1]See [110] (pp. 383-388) for a formal proof that an encryption scheme is semantically secure if and only if it has ciphertext indistinguishability.

First we let a polynomial-size circuit family denoted $\{C_n\}^\infty$ be defined as an infinite sequence of Boolean circuits $C_1, C_2, \ldots$ such that for every n, the circuit C_n has n input nodes and size $p(n)$, where $p(\cdot)$ is a polynomial that is fixed for the entire circuit family. Independent of $p(n)$, we also let poly(n) denote an arbitrary, unspecified polynomial in n which is used to bound the length of plaintexts. The ciphertext indistinguishability of a symmetric encryption scheme is then defined as follows:

Definition 2.2.2 (Ciphertext Indistinguishability) A symmetric encryption scheme (G, E, D) has ciphertext indistinguishability if for every polynomial-size circuit family $\{C_n\}^\infty$, every positive polynomial p, all sufficiently large security parameters n and every pair of plaintexts $(x, y) \in \{0,1\}^{\text{poly}(n)}$:

$$Pr[C_n(E_{G(1^n)}(x)) = 1] - Pr[C_n(E_{G(1^n)}(y)) = 1] < \frac{1}{p(n)}$$

For a complete model of a cryptosystem which can be used to reason about high-level security properties such as confidentiality or authentication, it is necessary to contextualise computational complexity definitions in terms of an attacker model. There are four main attacker models which are used to specify the power of the adversary in relation to the underlying encryption scheme:

Ciphertext-only attack The adversary is passive, learns a ciphertext and then attempts to derive the plaintext. This is the least powerful attacker model.

Known-plaintext attack This models a more persistent, but still passive, adversary who learns a number of corresponding plaintext and ciphertext pairs. The adversary also learns an additional target ciphertext and attempts to determine the plaintext.

Chosen-plaintext attack The adversary this time is active, and may request the encryption of an arbitrary number of plaintexts. Afterwards, the adversary learns an

additional target ciphertext and attempts to determine the plaintext.

Chosen-ciphertext attack This models a powerful active adversary who, in addition to learning the encryption of arbitrary plaintexts, learns the decryption of arbitrary ciphertexts. Afterwards, the adversary learns an additional target ciphertext and attempts to determine the plaintext.

Modern ciphers aim to provide ciphertext indistinguishability under the chosen plaintext attacker model (IND-CPA). This is a conservative approach based on the fact that an encryption scheme that is chosen-plaintext secure is also guaranteed to be secure against passive, ciphertext-only and known-plaintext adversaries.

IND-CPA for private-key encryption schemes is typically formalised, for example by Katz and Lindell [147], using a game-based model in which an adversary $\mathcal{A}$ is allowed to ask for encryptions of multiple messages chosen adaptively. In particular, $\mathcal{A}$ can do polynomially-bound private computations and is allowed to interact freely with a challenger $\mathcal{C}$ that provides access to an encryption oracle. Each time the adversary $\mathcal{A}$ submits a plaintext message m, $\mathcal{C}$ returns the corresponding ciphertext $c = E(k, m)$. The IND-CPA game takes as input the security parameter n, an adversary $\mathcal{A}$ and is defined as follows:

IND-CPA$^{\mathsf{PrivK}}_{\mathcal{C},\Pi}(1^n, \mathcal{A})$:

1. The challenger $\mathcal{C}$ generates a key k by simulating the key generation algorithm $G(1^n)$.
2. The adversary $\mathcal{A}$ is given the input 1^n and is allowed to submit a polynomial number of messages m to the challenger $\mathcal{C}$. Each message m is encrypted by $\mathcal{C}$ and the corresponding ciphertext $c = E(k, m)$ is given to $\mathcal{A}$. Eventually, $\mathcal{A}$ submits a pair of equal-length messages (m_0, m_1) to $\mathcal{C}$.
3. $\mathcal{C}$ draws a random bit $b \in \{0, 1\}$ and then returns the encryption of message m_b which is called the challenge ciphertext $c^\star = E(k, m_b)$.
4. Once more, the adversary $\mathcal{A}$ is allowed to submit a polynomial number of messages

m to $\mathcal{C}$ and receives the corresponding ciphertexts.

5. Eventually, $\mathcal{A}$ outputs a bit b' indicating which message $m_{b'} \in (m_0, m_1)$ it believes $c^\star$ is an encryption of.
6. The output of the game is defined to be 1 if $b' = b$ and 0 otherwise.

Let a negligible function be defined as one that gets smaller faster than $\frac{1}{\mathsf{poly}(n)}$ for any polynomial poly in the security parameter n, then IND-CPA security for private-key encryption schemes is defined as follows:

Definition 2.2.3 (IND-CPA secure) A private-key encryption scheme $\Pi = (G, E, D)$ has indistinguishable encryptions under chosen-plaintext attack, also known as IND-CPA secure, if for all probabilistic polynomial time adversaries $\mathcal{A}$ there exists a negligible function negl such that:

$$Pr[\mathbf{IND\text{-}CPA}^{\mathsf{PrivK}}_{\mathcal{C},\Pi}(n) = 1] \leqslant \frac{1}{2} + \mathsf{negl}(n)$$

The formal security paradigm is an important methodology for evaluating the security of cryptographic algorithms based on assumptions about the computational abilities of an adversary. Game-based security definitions provide a convenient way of proving the security of a cryptosystem based on some underlying computational assumption or theory.

Whilst most public-key cryptosystems can be proven secure based on a reduction in their security to the hardness of some computationally difficult problem such as integer factorisation, constructing efficient symmetric-key schemes based on these assumptions remains an open problem. Instead, the security of private-key algorithms is typically found in their resistance to cryptanalysis. A successful cryptanalyst may break the semantic security of a symmetric encryption scheme by finding some advantage in determining the plaintext from the ciphertext based on flaws in the design of the cipher.

2.2.1 Kerckhoffs' Principle

Embedded in the private-key encryption setting is the idea that the adversary Eve knows everything that Alice and Bob do, except for the key. In 1883, long before the development of modern cryptography, Auguste Kerckhoffs made a lasting contribution to the art of cryptography in his study of design principles for military ciphers [149]. In what is now known as Kerckhoffs' Principle, Kerckhoffs wrote the following:

> "Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi."

In other words, the encryption, decryption and key generation algorithms which constitute an encryption scheme should not be kept secret. The confidentiality of the information being protected should rest only in the secrecy of the key. Despite Kerckhoffs' prescience, many modern systems have been developed using a "security by obscurity" approach. The automotive industry in particular developed a number of closed-source, proprietary cryptosystems which have all been proven to be insecure once the details of the algorithms were revealed [126, 105, 249, 248, 28].

2.2.2 The Key Distribution Problem

A symmetric encryption scheme provides a mechanism for ensuring the confidentiality of data that is sent over an insecure channel, but requires the establishment of a confidential key which is shared between the sender and receiver. The difficulty of establishing such a key is known as the key distribution problem. Besides physically distributing keys in advance of their requirement using an out of band channel, there are a two main techniques which address the key distribution problem. The first approach depends on a security protocol and the availability of a trusted third party to help generate and distribute keys. The Needham-Schroeder [178] and Kerberos [225] protocols are notable examples of

methods which allow a secret key to be established between two parties at the time they need to communicate.

The second technique for addressing the key distribution problem arises from a landmark discovery in the history of cryptography. First presented by Whitfield Diffie and Martin E. Hellman in 1978 [68], public-key cryptography transformed modern cryptography by introducing the idea that different keys could be used for encryption and decryption. With public-key cryptography, Alice and Bob make their encryption keys publicly available and no-longer need to establish a shared key in advance of their secure communication. Diffie and Hellman's eponymous key agreement protocol, still in widespread use today, provides a way for two parties to establish a secret key over an insecure channel without the involvement of a third party. Realising a public-key cryptosystem requires a certain amount of additional structure that makes them too slow, even today, for high-throughput applications. Instead, public-key cryptography is often applied only initially, for key agreement, and then a symmetric encryption scheme is used for encryption [170].

2.2.3 Key Length

The set of all possible keys for a cryptosystem is called the key space. The number of different keys is an important measure because one strategy for an adversary is to exhaustively try to decrypt a ciphertext using every key in the system. Since modern cryptosystems tend to have extraordinarily large key spaces, we typically measure their resistance to exhaustive search in terms of the key size or length instead [170]. The key length of a symmetric encryption scheme is usually expressed as the number of bits which are required to express the largest possible key value, so a symmetric cryptosystem with a k bit key length will generally have a key space of 2^k many keys. Today, systems which aim to provide data confidentiality during the next 10 years are advised to use a symmetric key length of no less than 128 bits [6].

2.3 A Taxonomy of Symmetric Encryption

There is enormous diversity in the construction of algorithms for providing symmetric encryption. One high-level way of categorising ciphers is as either stream or block ciphers depending on the number of bits which they operate upon [185]. Stream ciphers typically process plaintext one bit at a time and, correspondingly, output one bit of ciphertext. In contrast, block ciphers operate upon blocks of plaintext, typically 64 or 128 bits, and output blocks of ciphertext. Traditionally, stream ciphers have been smaller and more efficient than block ciphers [185] and have therefore been popular in embedded applications. In modern practice, most symmetric encryption is performed using the AES block cipher.

2.3.1 Stream Ciphers

A stream cipher, as shown in Figure 2.3, encrypts bits individually by adding, modulo 2, each bit in the plaintext to one bit from the keystream. The security of a stream cipher depends entirely on the keystream [185] and so the design of stream ciphers is primarily concerned with the generation of a good, unpredictable keystream. Formally, the keystream should be a Cryptographically Secure Pseudorandom Number Generator (CSPRNG) that creates sequences of bits $s_1, s_2, \ldots, s_n$ such that its computationally infeasible to compute the bits $s_{n+1}, s_{n+2}, \ldots$ [109].

Feedback Shift Registers

Feedback shift registers, and in particular Linear Feedback Shift Registers (LFSRs) as shown in Figure 2.4, are well suited to low-cost hardware implementation and are a very common method for providing the long pseudorandom sequences required by stream ciphers [185]. Whilst LFSRs have good statistical properties they do not have good cryptographic properties and many prominent LFSR-based ciphers such as A5/1, used for encrypting GSM cellular telephone communications, have been found to contain critical flaws in their

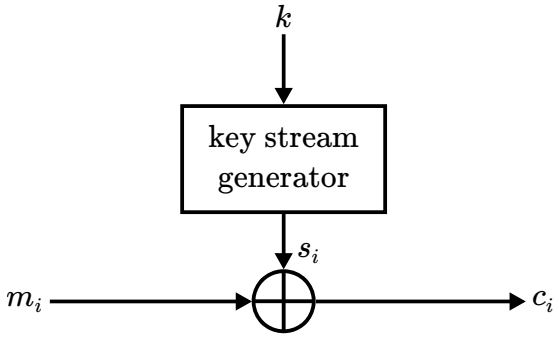


Figure 2.3: A stream cipher that takes as input a plaintext bit m_i and a key k. The ciphertext output bit c_i is computed by adding modulo 2 the keystream bit s_i and the input bit m_i.

design [112, 124, 180]. LFSRs are cryptographically troublesome because their input bit is a linear function of the register's previous state. This means that the output from a LFSR will always reveal some information about the internal state of the register at a previous time. Cryptographically this means that the output from a LFSR can often be used to determine some bits from key that was used to initialise the register. A brief taxonomy and mathematical treatment of LFSRs is given in [185].

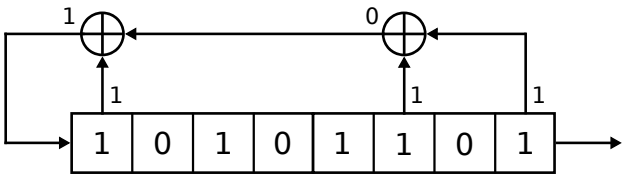


Figure 2.4: LFSR with bits $0, 5$ and 7 tapped. On the next clock cycle, this LFSR will output the bit 1 and the input bit will also have value 1.

Non-Linear Feedback Shift Registers (NLFSRs) are a generalisation of LFSRs in which the input bit is computed using a non-linear function of the previous register state [145]. Although NLFSRs have a much higher linear complexity than LFSRs of the same order, the theory of their operation is less complete and many open problems in their systematic construction remain [75]. Despite most early shift-register based encryption algorithms being broken, the European Network of Excellence in Cryptography (ECRYPT) eSTREAM

project has since developed a modern portfolio of secure and efficient stream ciphers based on LFSRs and NLFSRs that are suitable for use in new designs [12].

2.3.2 Block Ciphers

A block cipher, as shown in Figure 2.5, encrypts an entire block of plaintext bits at the same time, using the same key. The number of bits that a block cipher operates on is called its block size, and is typically 128 bits in modern algorithms. Claude Shannon, in addition to his seminal work on information-theoretically secure encryption which we introduce in Section 2.2, developed several techniques for constructing concise, pseudorandom permutations [216]. Shannon identified two primitive encryption operations, confusion and diffusion, that both frustrate statistical analysis. Shannon also recognised that concatenating many individually weak operations together could yield a collectively secure construction. Shannon's technique, the repeated application of confusion and diffusion, is at the heart of all modern block ciphers [185]. Multiple confusion and diffusion primitives are arranged into rounds and the encryption of one plaintext block comprises the execution of many rounds. Typically, each round uses a unique key which is derived from the main encryption key using a key schedule. The security of a block cipher, specified in terms of its block size and key length as well as the number of rounds, is measured in terms of ciphertext indistinguishability as given in Definition 2.2.2.

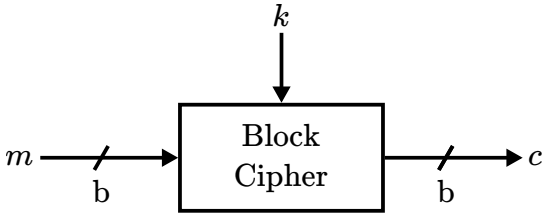


Figure 2.5: A block cipher that takes as input a plaintext m of size b bits, a key k and outputs the ciphertext c, also of size b bits.

Feistel Networks

Although now superseded by AES for all but legacy applications, one of the most influential and widely used block ciphers in the last 30 years is DES [185]. DES, a U.S. government approved standard for encryption between 1977 [63] and 2005 [65], was based on the design of an earlier cipher developed by IBM's cryptography research group during the late 1960s [61]. Horst Feistel, who headed IBM's research group at the time, developed the Lucifer cipher [91] that provided the basis of DES and which is also one of the easiest civilian block cipher designs [90]. Both Lucifer and DES are based on a general method of transforming any function into a permutation called a Feistel network. Feistel networks have since been used in the design of many block ciphers including FEAL [217], GOST [266], Khufu and Khafre [175], Blowfish [209], RC5 [205], MISTY [172], PRESENT [25], SIMON and SPECK [13].

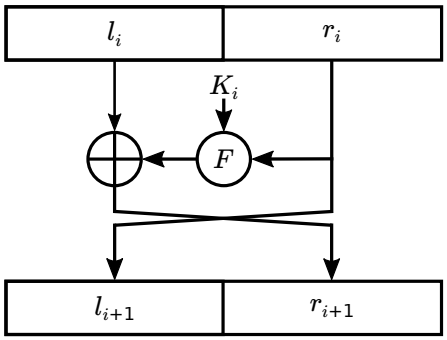


Figure 2.6: A balanced Feistel network construction. In each round i, the input string is divided equally into its left l_i (target) and right r_i (source) halves and a round key K_i is derived from the key K according to a key schedule. The round output is $r_i \parallel l_i \oplus F(K_i, r_i)$.

The fundamental building block of a Feistel network is the key dependent Feistel function that maps an input string onto an output string. The Feistel function is always non-linear and is often irreversible. The security of a Feistel network is based on the iteration of the Feistel function and so the number of rounds required for resistance to a given attack is dependent on the properties of the function [211].

Conventionally, Feistel networks are balanced and, as shown in Figure 2.6, the Feistel function is applied to exactly half of the input block in each round. In a balanced Feistel network the right half of the input is termed the source block and the left half is termed the target block. Feistel networks may also be unbalanced in which case the input block is divided arbitrarily between the source and the target block. In this thesis we adopt the terminology of Schneier and Kelsey [211] for describing unbalanced Feistel networks. In particular, the only requirement for a construction to be classified as a Feistel network is that one part of the block being encrypted influences the encryption of another part of the block. Whilst most Feistel networks are balanced, MD5, Khufu and Khafre and all NLFSRs are examples of unbalanced Feistel networks.

2.4 Key Management

The secure generation, distribution, storage, use and destruction of cryptographic keys is the domain of *key management*. Good key management is critical to the security of any cryptosystem, which cannot be secure if the key is known or can be derived by an adversary. The Wired Equivalent Privacy (WEP) wireless access protocol is an example of a high profile cryptographic implementation which was compromised because of poor key generation techniques [226]. Bad key management in WEP ultimately reduced the security afforded by a 104 bit key to that of resisting an active attacker for fewer than 60 seconds [234]. The automotive industry has been particularly guilty of bad key management, often neglecting to make any attempt whatsoever to ensure the secure generation or distribution of cryptographic keys [105]. As recently as late 2018, Tesla's high-end Model S was found to be using a 40 bit symmetric encryption scheme to secure the Remote Keyless Entry (RKE) system on the vehicle. Attackers were able to clone a key fob using off-the-shelf equipment in seconds, allowing them to unlock and start the vehicle at any time [264]. The same weakly keyed encryption algorithm, DST40, was already known to be insecure

since 2005 when it was first discovered in a vehicle immobiliser system deployed in millions of vehicles [28].

2.5 Vehicle-to-Everything Communication

In the near future, vehicles will communicate directly between themselves and with roadside infrastructure. V2X communication, which includes Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) techniques, is expected to drastically improve road safety and efficiency by enabling the next generation of semi-autonomous vehicle safety features such as vehicle platooning, collaborative forward collision warning and emergency electronic brake lights [246]. The systems which will provide safety and efficiency features based on V2X communication are termed Intelligent Transportation Systems (ITS) [72]. The vehicles and roadside infrastructure which participate in ITS are collectively termed ITS-Stations (ITS-S).

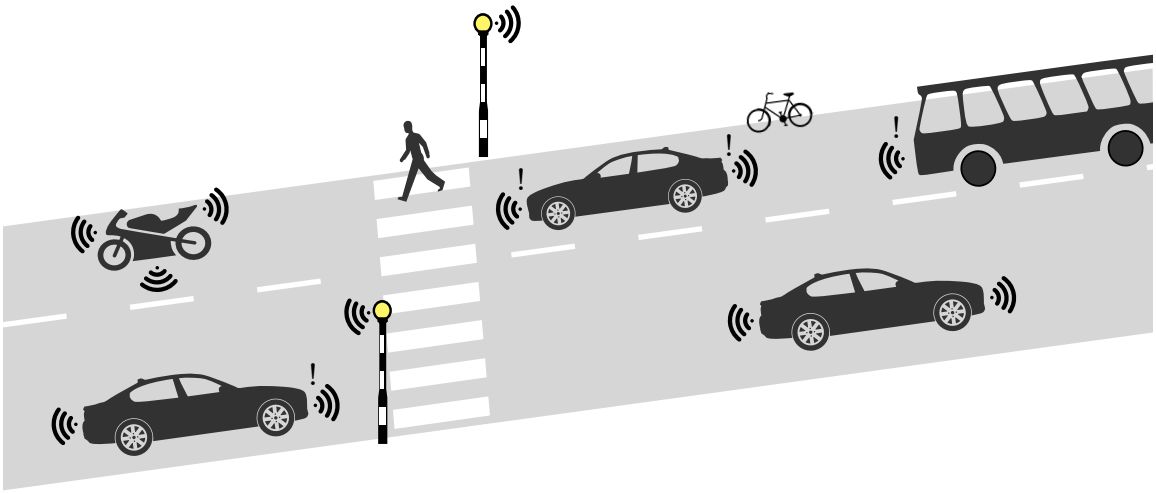


Figure 2.7: Vehicle-to-Everything Communication.

As illustrated in Figure 2.7, V2X enables the spontaneous creation of Vehicular Ad-hoc Networks (VANETs) which are used to provide each vehicle with an enhanced situational awareness. Vehicles endowed with a more detailed and far reaching awareness of their environment are able to operate more safely by avoiding taking actions that might cause a collision. The inclusion of infrastructure such as traffic signals, cameras and power-grid terminals enables the development of ITS which realise new efficiencies in road and power

usage. Traffic data can be aggregated from roadside infrastructure and used to provide vehicles with routes that minimise congestion [254] and balance the load on the power grid [221].

Vehicular safety features, including V2X, are most effective when they are universally deployed [138]. As such, there is a need for common V2X standards which will allow for a coordinated and internationally interoperable deployment. There are a number of key V2X standards which are supported by both European and U.S. governments [72, 53] and which have been adopted for imminent deployment by industry [156, 253].

2.5.1 WLAN vs. Cellular communication for V2X

V2X calls for a reliable, low-latency wireless communication technology which allows vehicles to send and receive safety-critical messages in a highly dynamic environment. Traditionally, a Wireless Local Area Network (WLAN) based technology has been chosen for this role. As early as 1999, the U.S. Federal Communications Commission (FCC) allocated a portion of the 5.9 Ghz electromagnetic spectrum for ITS-enabling Direct Short Range Communication (DSRC) between vehicles [89]. Today, the international physical-layer wireless communication standard for V2X is IEEE 802.11p [206]. IEEE 802.11p provides the amendments to the IEEE 802.11 Wi-Fi standard that are necessary to provide wireless communications in a vehicular environment [144]. IEEE 802.11p is part of the IEEE Wireless Access in Vehicular Environments (WAVE) standards which also define an architecture [130], services [132] and message format [131] for DSRC. The European equivalent of DSRC is termed ITS-G5 [82] and is standardised by the European Telecommunications Standards Institute (ETSI). Both ITS-G5 and DSRC use IEEE 802.11p wireless communication and the IEEE WAVE message encoding.

The use of ITS-G5 is strongly encouraged in the EU by a mandate which requires member states to adopt an interoperable standard for V2X [72]. In America, the U.S.

Department Of Transportation (U.S. DOT) have pushed for the legislation [96] and accelerated utilisation of DSRC [53].

Recently, proponents of using cellular technology for V2X applications have emerged. The 3rd Generation Partnership Project (3GPP) is a standards organisation which develops the industry standards for mobile communication such as Fourth-Generation Long-Term Evolution (4G LTE) and Fifth-Generation (5G) cellular network technology. In 2016, 3GPP completed its initial standard for cellular V2X [137]. The emergence of a competing technology for V2X after nearly two decades of research, standardisation and legislation has caused a fierce debate to arise, with invested parties on both sides making the case for their technology [239, 53, 36, 93]. In spite of the ongoing battle to establish a single dominant technology for V2X, it has been suggested that IEEE802.11p WLAN and 5G are actually complementary technologies which can be harmonised for ITS [94]. In a hybrid communication model, 802.11p WLAN technologies would be used for safety-critical V2V applications and 5G would be used for less latency sensitive cloud based applications [1].

2.5.2 Cooperative Awareness Messages

In V2X, the safety-critical messages that are sent between vehicles are termed either Cooperative Awareness Messages (CAM) [80] or Basic Safety Messages (BSM) [207]. CAM are broadcast from vehicles to nearby receivers with a high frequency, typically 10 times per second. As shown in Figure 2.8, a CAM contains a protocol data unit (PDU) header, a basic container and a high frequency container. Conditionally, a CAM may also contain a low frequency container and special vehicle container. CAM facilitate the most imminent ITS service, cooperative awareness.

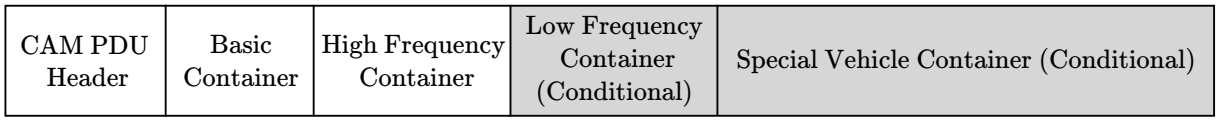


Figure 2.8: General Structure of CAM.

The CAM basic container comprises the type (e.g. passenger car, bus, pedestrian or road side unit) and the latest geographic positioning of the originating vehicle. The high frequency container contains dynamic status information such as heading and speed and the conditional low frequency container specifies static or slow changing information such as the status of exterior lights. The special vehicle container is reserved for road users which require special treatment such as emergency vehicles, vehicles containing heavy or dangerous loads and road works. CAM are specified fully in ETSI standard EN 302 637-2 [80].

In addition to cooperative awareness, the basic set of services for ITS also includes Decentralized Environmental Notification (DEN). DEN Messages (DENM) provide information related to events that have a potential impact on road safety and can also be used for traffic efficiency applications [81]. In contrast to CAM, DENM are less latency sensitive and may be disseminated over long distances.

2.5.3 A Secure On-board Vehicular Architecture

A typical modern vehicle is fitted with dozens of interconnected ECUs, forming a complex distributed system. The "E-safety Vehicle Intrusion Protected Applications" (EVITA) project [76] developed a secure on-board vehicle architecture specifically tailored to the security requirements of V2X communication. In the EVITA architecture, each vehicle ECU is combined with a Hardware Security Module (HSM) and the security functions of the vehicle are partitioned between hardware and software. In order to meet the cost-requirements for practical deployment, EVITA supports a broad range of secure hardware ranging from full external HSMs through to lightweight cryptographic coprocessors embedded into an ECU chip. Regardless of the implementation, the main idea is that each ECU includes tamper-resistant hardware that provides a root of trust, secure key storage and can perform basic cryptographic operations.

The assumption of trusted hardware on-board the vehicle is made by both of the leading V2X standards. Each vehicle is fitted with an On-Board Unit (OBU) which enables V2X communication. The OBU contains a tamper-resistant Trusted Element (TE) hardware device, such as a smartcard, which meets the requirements of the EVITA architecture.

2.5.4 Security, Privacy and Trust Requirements for V2X

It is imperative that V2X is designed to resist attacks and that the awareness provided by CAM can be trusted to direct safety-critical vehicle behaviour. It is also vitally important, and a legal requirement in Europe [183], that V2X functionality does not undermine or harm the privacy of road users.

The geographic positioning and dynamic status information broadcast by vehicles is fundamental to providing cooperative awareness as a service. The short-term linkability of vehicles in this context is a fundamental aim of V2X, without which vehicles could not accurately determine their environment. The open nature of V2X also means that adversaries may come from inside as well as outside the system. A V2X architecture must harmonise the functional requirement for close-range vehicle linkability with the need to adequately protect road users from the type of long-term tracking that threatens to uniquely identify individual habits. At the same time, a V2X architecture must provide a way to identify and exclude internal attackers who send misleading messages.

The European Preparing Secure Vehicle-to-X Communication Systems (PRESERVE) project [214], with input from ETSI and the Car-to-Car Communication Consortium (C2C-CC), identified many of the security, privacy and trust requirements for V2X around which the latest international standards were developed. The standard security, privacy and trust requirements for V2X are as follows:

Authentication and Authorisation The main security requirement for V2X is that the authenticity and integrity of CAM can be determined. If a vehicle is to take

safety-critical actions based on cooperative awareness, it must be able to trust that a received CAM was sent by a legitimate, non-revoked, ITS-S. In addition to authenticity and integrity, authorisation is required in order to restrict access to legitimate users.

Confidentiality As CAM are broadcast to all nearby receivers, there are no confidentiality requirements [86]. Confidentiality would risk potentially life-saving messages from being unreadable at the moment they were needed.

Privacy The two main privacy requirements for V2X communication are pseudonymity and unlinkability [87]. Pseudonymity allows each ITS-S to participate in cooperative awareness and other services without disclosing its identity but still ensures that it can be held accountable for its usage. Unlinkability ensures that an ITS-S can repeatedly use a service or resource without an observer being able link the usage to a single source. Unlinkability and pseudonymity are required both against passive inter-vehicle adversaries which observe broadcast messages and against the infrastructure that manages trust in ITS.

Sybil Attack Resistance Related to the abuse of anonymous and pseudonymous systems, V2X requires measures which limit the potential for an ITS-S to execute a Sybil attack [73] in which a large number of other vehicles or roadside units are imitated.

Removal of Misbehaving Vehicles Related to the trust in CAM, it is essential that compromised ITS-S that systematically send erroneous messages can be removed from participation.

Optional Law Enforcement Support An optional requirement for ITS, that may further improve trust in CAM, is the provision of a controlled de-pseudonymisation

mechanism. Allowing a suitable authority to selectively de-pseudonymise specific CAM would deter both electronic misuse and vehicle misbehaviour.

2.5.5 Performance Requirements for V2X

In addition to the security, privacy and trust requirements for V2X, there are a number of performance requirements that are based on the anticipated computational and bandwidth limitations of first and second-generation connected vehicles. The standard performance requirements for V2X are as follows:

CAM Processing Throughput The PRESERVE project [214] established the requirement for vehicles to process at least 1,000 CAM per second. Given the hardware standards of the time, PRESERVE identified that first generation ITS-S would require a cryptographic coprocessor to achieve the necessary throughput.

Efficient Revocation Processing a CAM includes ensuring that the ITS-S which sent the message has not been revoked for misbehaviour. Given the anticipated scale of ITS and the already tight processing constraints, only very efficient revocation mechanisms are suitable for V2X.

Limited OBU Storage Requirements The effectiveness of V2X for safety and efficiency is proportional to the number of vehicles that support the technique. To support the broadest possible range of vehicles, V2X schemes are required to minimise the on-board storage that they require.

TE Simplicity The TE or HSM that provides cryptographic functionality within an ITS-S [76] should be simple so that auditing can provide a high-assurance of its exact behaviour. A V2X scheme should minimise the required TE functionality so that a high degree of trust can be established. Keeping the required TE functionality

limited also minimises cost and enables off-the-shelf devices such as smartcards to be used.

Limited Bandwidth and Intermittent Connectivity As a safety-critical system, V2X is required to operate despite limited bandwidth and intermittent internet connectivity. Bandwidth usage should be minimised and V2X must operate without access to online infrastructure.

2.5.6 PKI for V2X

In the V2X communication model, vehicles form spontaneous cooperative networks between themselves and with roadside infrastructure. Because of the topology and scale of V2X networks, using pre-shared symmetric keys is not suitable and a public-key solution is required. Whilst public-key cryptography solves the key distribution problem, it introduces the related issue of how to bind a public key to a specific identity. The most common method for binding a public key to a specific identity is through the use of public-key certificates, which provide a mechanism for providing assurance about the purpose of a public key [170]. Key management for public-key cryptosystems, typically by supporting public-key certificates, is the domain of Public Key Infrastructure (PKI).

PKI for V2X is primarily required to support the issuance and provision of public-key certificates to vehicles and roadside infrastructure, to limit the misuse of credentials by controlling the validity of issued certificates and to exclude compromised or misbehaving entities by revoking their credentials [23]. There are two leading PKI proposals for V2X, the European ETSI approach [86] and the American USDOT Security Credential Management System (SCMS) [258, 29]. Although the terminology differs slightly, both proposals include the following key entities:

Root Authority The role of a Root Certificate Authority (RCA) is to define a common policy which is inherited by the subordinate certificate authorities. The RCA issues

long-term certificates to the enrolment and authorisation authorities.

Enrolment Authority An Enrolment Authority (EA) is responsible for the issuance and provision of long term vehicle credentials. The EA will typically issue a pseudonymous vehicle certificate during manufacture and then maintain an internal record linking the certificate to a specific vehicle identity.

Authorisation Authority The role of an Authorisation Authority (AA) is to authorise, through the provision of tickets, an ITS-S to use a particular application, service or privilege [85]. An ITS-S will request access to a particular service by presenting its long term enrolment credential. If the ITS-S is authorised then the AA will respond to the request with a pseudonymous authorisation ticket which specifies a set of permissions. In terms of the cooperative awareness service, the permissions on a ticket relate to the time period and message set for which the ticket is valid.

The ETSI standard PKI architecture [87] incorporating these roles is shown in Figure 2.9. Both the ETSI architecture and the USDOT SCMS [29] use Elliptic Curve Digital Signature Algorithm (ECDSA) [70] signatures to provide authentication and authorisation. ECDSA is the prevailing signature scheme for V2X because it provides the small signature size and low computational complexity that are required [214]. ECDSA is used for the internal certification of authorities, enrolment credentials, authorisation tickets and for signing the CAM that are sent between vehicles.

For privacy, both standards use the role-separation of certificate authorities. Since the canonical ITS-S identifier is only known to the EA, and a pseudonymous enrolment credential is used to request authorisation tickets from the AA, the AA cannot link pseudonyms to a canonical ITS-S identity. The unlinkability of communications between ITS-S is provided by the issuance of time-limited authorisation tickets which each provide a different ITS-S pseudonym. The task of deciding when to change pseudonyms for optimal

privacy remains an open problem and neither standard yet agrees on a common strategy for pseudonym change [84].

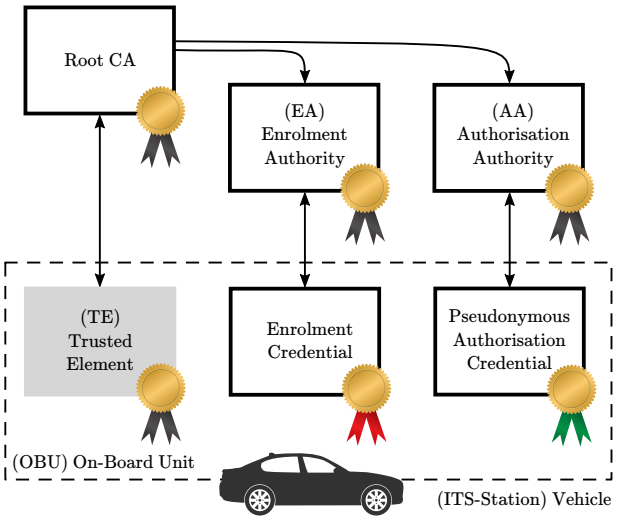


Figure 2.9: ETSI Standard V2X PKI model.

The ETSI and USDOT standards take different approaches towards the removal of misbehaving vehicles. One of the most common methods for revocation is the use of Certificate Revocation Lists (CRLs). Given the anticipated scale of ITS and the tight computational constraints on CAM processing, ETSI do not consider CRL viable and prefer passive revocation [87] through the issuance of time-limited authorisation tickets that are renewed frequently. Compromised vehicles are simply denied additional pseudonyms and become unable to participate. The USDOT SCMS uses a different approach based on implicit certificates [33] and a 'butterfly key expansion technique' which enable a more efficient use of CRL.

Chapter 3

Related & Previous Work

In the previous chapter, we developed an understanding of the basic methods of cryptographic analysis for electronic immobiliser systems and introduced the foundations of V2X communication. In this chapter, we present the literature that builds upon these techniques and that relates to the development and evaluation of electronic security and safety systems for vehicles. We introduce complementary studies relating to key management for public-key cryptosystems and to the design of secure and privacy-friendly protocols and architectures based on the utilisation of trusted hardware.

In addition to the literature that directly relates to the analysis of electronic vehicle immobiliser systems, we also introduce the more general body of work that is based on the (in)security of proprietary automotive cryptography. In particular, many of the techniques developed in the cryptanalysis of keyless entry systems are highly applicable. We then position the security of these systems in the context of the modern automotive attack surface.

Relating to electronic safety systems for vehicles, we focus on methods and architectures for secure V2X communication. We consider the latest standards for V2X and the body of work that accompanied to their development, then broaden our discussion to include notable techniques from the wider literature.

3.1 Proprietary Automotive Cryptography

For many years, only weak, proprietary cryptography was implemented in electronic automotive systems worldwide [105]. In this section, we first introduce the literature on vehicle immobiliser systems and then we extend our discussion to include a broader range of automotive and proprietary cryptographic systems.

3.1.1 Electronic Immobiliser Systems

Automotive cryptography first received attention from the academic community in 2005, when Bono et al. [28] reverse engineered Texas Instrument's Digital Signature Transponder (DST) immobiliser system and found that it was insecure. The authors discovered that DST, which at the time was being used to secure 7 million electronic payment devices in addition to millions of vehicles, was based on the security of the eponymous DST-40 cryptographic encryption algorithm. DST-40, also known as the Kaiser cipher [114], is a 200 round unbalanced Feistel network cipher that takes as input a 40 bit secret key and has a block size of 40 bits. The key schedule is a simple LFSR that provides a new round key every 3 rounds.

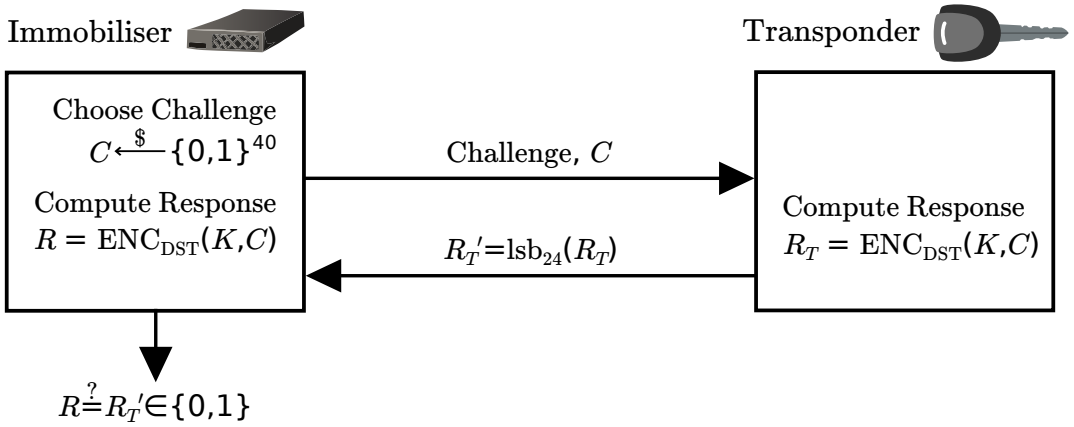


Figure 3.1: The prototypical DST-40 challenge-response authentication protocol.

The DST immobiliser system implements a basic challenge-response protocol that

is typical of an electronic vehicle immobiliser. As shown in Figure 3.1, the immobiliser sends a 40 bit challenge nonce to the transponder and then locally encrypts the challenge using the shared symmetric key, so that the transponder's response can be verified. The transponder encrypts the challenge using the DST-40 algorithm and then sends the lower 24 bits of the encrypted challenge back to the immobiliser. On average, only a transponder with the matching 40 bit secret key will provide the correct response and so DST provides a mechanism for authenticating the transponder to the vehicle immobiliser system.

Bono et al. [28] reverse engineered the DST-40 implementation by following a black-box approach in which they uncovered the functional details of the cipher by examining the logical outputs of a DST device. The authors then built a dedicated DST-40 key cracking device for $3,500 that enabled them to recover five DST keys in less than two hours. Ultimately, any application using DST-40 is undermined by the inadequacy of a 40 bit secret key in resisting a modern exhaustive key search attack [6].

The seminal work on DST-40 was not only the first to identify critical flaws in a widely used vehicle immobiliser system, and lay many of the foundations for recovering and analysing proprietary automotive cryptography, but also represents one of the first published attacks on a commercial device in the literature [105]. In light of their findings, the authors urged automotive system designers to embrace standard cryptographic algorithms such as 128 bit AES [5] or Keyed-Hash Message Authentication Code (HMAC) [235]. Despite this advice, DST-40 resurfaced again in late 2018 when Tesla's high-end £75,000 Model S was found to be using the algorithm to protect its RKE system. This time attackers were able to clone a key fob in seconds, allowing them to unlock and start the vehicle at any time [264].

Keeloq

DST-40 alerted the cryptographic research community to the fact that insecure, proprietary cryptography was being widely used in wireless consumer devices. It was not long before

the security of several other high-profile automotive systems came under scrutiny. In 2007, Andrey Bogdanov published the first cryptanalysis of Microchip's popular Keeloq cipher [26]. Keeloq is a NLFSR-based block cipher with a 32 bit block size and a 64 bit key length that was designed to facilitate very low cost and efficient hardware implementations. Microchip recommended the Keeloq algorithm for a broad range of applications including automotive immobilisers, access control systems, electronic door locks and burglar alarms [148]. Bogdanov's initial cryptanalysis identified a slide attack [20, 21] requiring 2^{32} plaintext-ciphertext pairs and that recovers the 64 bit Keeloq secret key after $2^{50.6}$ encryptions. An improved algebraic slide attack on Keeloq requiring only 2^{48} encryptions was published by Courtois et al. [57] and then Indesteege et al. [136] developed a novel meet-in-the-middle attack [69] requiring just 2^{16} plaintext-ciphertext pairs and $2^{44.5}$ encryptions. Finally, in [78, 146, 184] several hardware attacks were proposed that enable a Keeloq transponder to be cloned in minutes.

Hitag2

At the same time as the first attacks on Keeloq were published, an implementation of NXP's Hitag2 cipher was anonymously posted online [128]. Hitag2, introduced in 1996, is one of the most widely deployed vehicle immobiliser systems and is known to be used by at least 200 different vehicle models from more than 33 different brands [248]. Hitag2 is a stream cipher with a 48 bit key length that is based on the design of an earlier cipher, Crypto1 [104], used in the MIFARE Classic contactless smart card. Hitag2 is a simple algorithm that comprises a 48-bit LFSR and a non-linear filter function. For each clock tick, the filter function generates one keystream bit using 20 bits from the LFSR. Despite having improved security over the Crypto1 cipher, Hitag2 was broken in 2009 by Courtois et al. [58] who developed a generic algebraic attack able to fully recover the full 48 bit Hitag2 key in just a few hours. Further attacks on the Hitag2 cipher are given in [220, 270, 134] and then, in 2011, Sun et al. present a theoretical cube attack [71] that

can recover the secret key in less than a minute under chosen initialisation vctors [227]. In 2012, Verdult et al. [248] showed how several vulnerabilities in the Hitag2 immobiliser system can be combined to develop an attack that recovers the secret key after just 136 partial authentications with the vehicle immobiliser and 2^{35} operations. The attack takes less than one minute of communication with the immobiliser and around five minutes of offline computation using a standard laptop.

Megamos Crypto

In practice, almost all vehicles sold in Europe between 1995 and 2015 are fitted with an immobiliser system based on either the Hitag2 or Megamos Crypto encryption algorithm [247]. The Megamos Crypto algorithm was developed by Thales, who then licensed the design to EM Microelectronic. EM kept the details of the underlying cryptographic algorithm a secret whilst selling two different transponder packages containing the algorithm [59, 60]. Full details of the Megamos Crypto immobiliser system including the design of the cipher, its associated authentication protocol and several practical attacks on the system were first published by Verdult et al. [249] in 2013. Despite the authors responsible disclosure to EM Microelectronic in November 2012, Volkswagen (VW) Group, a key stakeholder with millions of affected vehicles, successfully applied to the High Court of Justice in the United Kingdom for an injunction that prevented key sections of the paper from being released [252]. When the full details were eventually published in 2015 [150], it was revealed that the Megamos Crypto algorithm is a block cipher with a 96 bit key length that comprises a LFSR, a NLFSR and three 7 bit registers. The authors propose several attacks. In the first, an adversary requires only two eavesdropped authentication traces and is able to recover the 96 bit secret key with a computational complexity of 2^{56} cipher ticks. The second attack exploits a weakness in the transponder key-update mechanism, requires 3×216 authentication attempts with the transponder, and in practice allows an adversary to recover the cryptographic key in just 30 minutes. The authors also identify

several weak keys that can be recovered in minutes using a standard laptop.

3.1.2 Keyless Entry Systems

In addition to an electronic immobiliser transponder, the ignition keys of many modern vehicles include a second cryptographic system with an entirely different purpose. Just as mechanical vehicle immobilisers have been superseded by their electronic counterparts, RKE systems are the digital equivalent of a traditional vehicle ignition key. As shown in Figure 3.2, a typical modern vehicle key includes a RKE system, a Radio Frequency (RF) transmitter and a button that, when pressed, allows the user to remotely lock and unlock the vehicle. As well as RKE, some high-end vehicle ignition keys also include a Passive Keyless Entry and Start (PKES) function that, when bought within close range of the vehicle, unlocks the doors, deactivates the alarm, and enables the engine to start. Typically in PKES, the immobiliser system in the vehicle is used to trigger the transmission of a door opening signal that is sent from the transponder in the key over its RKE interface.

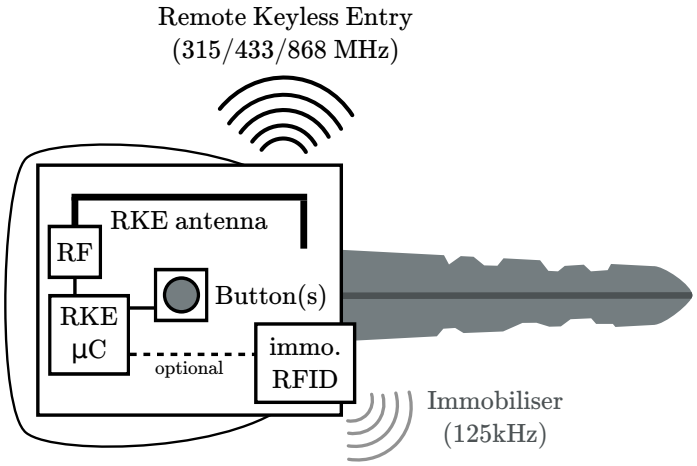


Figure 3.2: A typical modern vehicle ignition key comprising both a low-frequency immobiliser transponder and a high-frequency RKE transponder.

Passive Keyless Entry and Start

The security of PKES systems was first analysed by Francillon et al. [99] who demonstrated the susceptibility of vehicles from at least 8 different manufacturers to an adversary who relays the messages sent between the vehicle and the PKES transponder. The attack can be realised with as little as $100 dollars worth of equipment and allow an attacker to open and start a vehicle without physically compromising the key or raising any suspicion of the owner. Relay attacks are well known in the literature [120] and have also been shown on credit card transactions [74] and on proximity-type RFID tokens [119]. The authors propose that a RF distance bounding protocol such as [196], based on carefully measuring the round-trip time between cryptographic challenge and response, is the most appropriate long-term countermeasure against PKES relay attacks.

Remote Keyless Entry

As recently as 2000, some RKE systems used no cryptography whatsoever and simply transmitted a fixed code from the transponder to the vehicle [105]. Fixed-code RKE are vulnerable to straightforward replay attacks [229] in which an adversary just retransmits the authentication code that is sent by the transponder each time a user unlocks their vehicle. To address replay attacks, newer RKE systems implement cryptography and a rolling-code mechanism that provides synchronisation between a vehicle and its specific RKE transponder. In a rolling-code RKE system, both the vehicle and the transponder maintain a counter that keeps track of how many times vehicle has been successfully unlocked. Each time the button on the ignition key is pressed, the transponder counter value is incremented and then used to cryptographically generate the vehicle unlocking signal. Only if the unlocking signal is valid and the received counter value is greater than the vehicle counter does the RKE authentication succeed. Rolling-code based RKE systems prevent replay attacks because any previous unlocking signal that is retransmitted

will always contain an outdated counter value.

The insecurity of certain rolling-code based automotive RKE systems was first identified by Cesare in 2014 [46] who used phase space analysis [267] of the codes transmitted by an unidentified 2000-2005 model car to reveal that the codes required to unlock the vehicle were highly predictable. Shortly afterwards, a thorough case-study into the security of several market-leading RKE systems was given by Garcia et al. [105]. The authors identify four different RKE schemes used by VW and a fifth used by several manufacturers including Renault, Peugeot and Ford. The first VW scheme was found to be entirely non-cryptographic and trivially insecure. The second and third schemes used by VW revealed a proprietary cipher that was previously unknown in the literature, AUT64. Promisingly the fourth VW scheme was found to be using the open source XTEA block cipher [255, 179]. AUT64 and XTEA are both Feistel-network based block ciphers with 120 and 128 bit key lengths, respectively. The best known attack on XTEA [164] is still only theoretical, operates probabilistically on a reduced number of rounds, and reduces the security to $2^{104.33}$ encryption operations. Remarkably, the security of all three cryptographic VW RKE schemes is entirely undermined by the use of a single, global master key for each scheme. The authors discovered that virtually all VW group vehicles manufactured between 1996 and 2016 use one of three cryptographic master keys, without any key diversification whatsoever. These rolling-code based RKE schemes offer no cryptographic security at all and are entirely dependent on the *uid* of the transponder and the value of the rolling counter. In practice, these VW RKE schemes can all be broken by an adversary who eavesdrops one single unlock signal.

The fifth RKE scheme studied by Garcia et al. is a rolling-code based system that implements NXP's Hitag2 stream cipher. Hitag2 was already known to the academic community who had previously identified the algorithm and its weaknesses in a widely used immobiliser system [58]. Unlike the VW schemes, the Hitag2 RKE system does

not use a fixed global master key. Since the prior art on attacking Hitag2 [248] required too many authentication traces for use against an RKE system, the authors developed a novel correlation-based attack [52] that requires only 4-8 authentication traces and less than 10 minutes of computation using a standard laptop. Benadjila et al. [14] identify certain Hitag2 RKE systems that appear immune to the attack in [105] and propose an exhaustive search based technique requiring just 2 authentication traces and around 18 hours of computation using a single GPU. The authors point out that the same attack can be completed in 15 minutes using parallel GPU-based Amazon EC2 instances for a cost of around €45. Most recently, Verstegen et al. [251] develop a highly optimised guess-and-determine type attack [210, 247] against the Hitag2 RKE system that requires only 2 authentication traces and is able to recover the secret key in 75 seconds using a single GPU.

3.1.3 ECU Security

The immobiliser system in a vehicle is just one node in a large network of programmable microcontrollers and ECUs that collectively determine the functionality, safety and security of a modern vehicle. The most common method for interconnecting the ECUs in a vehicle is with one or more shared Controller Area Network (CAN) serial buses. The CAN bus standard [140], developed at a time when vehicles were essentially air-gapped systems [218], is not designed to withstand any adversarial behaviour. On one hand, this has meant that ECUs do not implement weak proprietary cryptography for intra-vehicle communication, something that seems likely given the historical automotive security landscape. On the other hand, the reason ECUs are not afflicted with proprietary cryptography for intra-vehicle communication is that they contain no cryptography of any kind and are entirely insecure against any passive or active adversary who can gain access to the CAN bus. As vehicles have become increasingly digital and connected,

the limitations of CAN and the traditional inter-vehicle adversarial model have become evident [155, 263, 262, 129, 152, 49]. In practice, an adversary who can compromise any single ECU onboard a vehicle is able to use the CAN bus to take complete control of the vehicle [152, 176]. In the context of a modern connected vehicle, the academic community has identified the need for cryptographic ECU source authentication and has proposed a number of solutions [195, 116, 242, 121].

Diagnostic Services

To allow the ECUs in a vehicle to be probed and updated there are a number of standard diagnostic protocols that are used during servicing. The most prevalent diagnostic protocols are the internationally standardised Unified Diagnostic Services (UDS) protocol [141] and its predecessor Keyword Protocol 2000 [142]. Koscher et al. [152] assess the security of ECU diagnostic services and identify a number of vulnerabilities that are made possible by weak or unenforced protection of reflashing capabilities. The authors uncover that a proprietary 16 bit challenge-response algorithm is used to protect the ECUs in their target vehicle from reflashing. The ECUs are required to allow one key attempt every 10 seconds and this allows an adversary to recover the secret key from any target ECU in around a week. The attack can be parallelised so that the secret keys for every ECU in a vehicle can also be recovered in the same time. Once several ECUs within the vehicle had been compromised and reflashed, Koscher et al. found they had almost complete freedom to tamper with safety-critical vehicle functionality. The authors were able to craft CAN packets that allowed them to arbitrarily lock and unlock the vehicle doors, turn off the power steering, disable the brakes and disable the engine. All of these attacks were possible despite the vehicle being driven at speeds of 40 mph and often it was not possible for the driver to manually override the attack.

Valasek and Miller [240] were the first to identify the use of proprietary cryptography for restricting access to ECU diagnostic services. The authors reverse engineered a

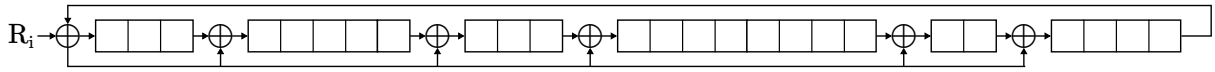


Figure 3.3: Structure of the proprietary Ford ECU access control cipher.

Ford diagnostic tool using the IDA Pro disassembler and uncovered a challenge-response algorithm that comprises a 24 bit LFSR based cipher. The cipher, shown in Figure 3.3, runs for 64 clock ticks during which a 24 bit challenge and a 40 bit secret key is shifted into the register. The response is computed as the key-independent permutation of the final 24 bit state of the LFSR. The authors did not analyse the cryptographic algorithm any further as they were also able to reverse engineer a hard-coded list of authentication keys that were present within the diagnostic tool. Valasek and Miller use the resulting elevated diagnostic privileges to demonstrate a number of safety-critical attacks, and then present data suggesting that a simplistic heuristic CAN bus analysis is sufficient to prevent many of the attacks they discuss.

Herrewegen and Garcia [241] reverse engineered 13 different ECUs from four different automotive manufacturers and identified three unique cryptographic algorithms that were being used for diagnostic access control. In addition to the Ford cipher already uncovered by Valasek and Miller [240], the authors also found a previously unknown Fiat cipher and another used by VW Group. The authors apply cryptanalysis to each of the algorithms, develop a number of practical key-recovery attacks, and then present a generic method for remotely executing code on compromised ECUs. To mitigate the vulnerabilities that their work exposes, Herrewegen and Garcia suggest a transition to standard public-key algorithms that would simultaneously allow for good key diversification between different ECUs, and that would also prevent an adversary from executing arbitrary code.

3.2 Vehicle-to-Everything Communication

In this section we introduce the most recent literature and standards relating to V2X communication and ITS. We begin with a brief history of the milestone V2X standards and research projects, before presenting the latest developments in security and privacy for vehicular networks.

3.2.1 A Brief History

In Europe, the €749 million PROMETHEUS project launched in 1987 [191, 260] aimed to develop new road traffic solutions that were safer, more efficient and better for the environment. One subprogram of PROMETHEUS in particular, PRO-NET, was specifically tasked with developing systems that enable vehicles to communicate with one another over data links [204]. Even at this early stage, the project anticipated that V2V communication would be applied to provide advance warning of obstacles or accidents and to enable vehicles to safely travel very closely together. Communication with roadside infrastructure was also considered by PROMETHEUS, under its PRO-ROAD subprogram that aimed to improve driver information by offering route guidance based on up-to-the-minute traffic information. The PRO-NET and PRO-ROAD subprograms were led by the automotive industry [4] and despite a number of practical demonstrations that established the technical feasibility of V2X communications [261, 43], neither program significantly contributed to the academic body of work on V2X.

There were a number of complementary early projects including DRIVE [203] in Europe and RACS and AMTICS [233] in Japan. In the U.S., the Intelligent Vehicle-Highway Systems (IVHS) Act of 1991 [139, 192] encouraged the development of a national strategy for ITS [228] that included the research, development, operational testing and deployment of V2X communication technologies. By 1993, the need for interoperable V2X was apparent and a number of international standards were correspondingly established [100].

In particular, the European Comité Européen de Normalisation (CEN) TC/278 standards committee formed in 1992 is still active today and works in collaboration with the International Organization for Standardisation (ISO) on a number of global cooperative-ITS standards [45]. From 1995 CEN took a leading role in the design of the 5.8 Ghz DSRC wireless standard [44] for V2X. An overview of the 1997 CEN DSRC draft standard is given by Detlefsen and Grabow [67]. Meanwhile, draft standards for DSRC were also prepared in Japan and by the American Society of Testing and Materials (ASTM) as surveyed by Yuan [265]. The ASTM draft version 7 introduced the 5.9 GHz frequency communication mechanism that is the basis of the latest IEEE 802.11p standard [206].

At the same time as the first draft standards for DSRC were published, preliminary results on the control system architectures necessary for vehicle platooning were demonstrated by the European Chaffeur project [107] and the California-based PATH Program [123]. In 2000, Tokuda et al. [236] developed a protocol for ad-hoc V2V communication based on 5.8 Ghz DSRC. The protocol was used to implement an autonomous driving demonstration that showcased 5 vehicles cooperatively engaging in platooning, lane changing, merging and obstacle detection and avoidance [238]. The demonstration was based on messages similar to ETSI CAM [80] and U.S. DOT BSM [207], that were transmitted 10 times per second and specified the speed and location of each vehicle.

3.2.2 Privacy

It is critical that ITS are designed to provide a high degree of user privacy. In Europe, citizens have a fundamental human right to protection with regards to the handling of their personal information [201]. The broadcast CAM that are the basis of V2X communication are personal data by definition and must be handled accordingly [183]. Defining privacy in V2X is challenging because the transmission of unencrypted messages that specify the precise location, speed and heading of each vehicle is a functional and safety-critical

system requirement [214]. The short-term precision and linkability of broadcast V2X messages is what enables road users to develop a more detailed and expansive model of their environment, so that their actions can be decided more intelligently and the number of accidents can be reduced.

Achievable Privacy

Before surveying the literature on privacy techniques for V2X, we briefly consider the scope of achievable privacy. The level of achievable privacy guides the creation of the adversarial model that is used to evaluate the performance of candidate ITS security architectures. Privacy in V2X has two main components. There is inter-vehicle privacy that is concerned with an adversary that intercepts the messages sent between road users, and there is also architecture privacy that is concerned with privileged adversaries that may reside within the infrastructure.

Defining achievable privacy in an inter-vehicle V2X setting is a challenging problem because there is a strong contention between the functional and privacy requirements. Providing accurate spatio-temporal information about the behaviour of road users is exactly what makes V2X a promising safety mechanism. At the same time, periodic positional information is inherently susceptible to techniques such as Multi Hypothesis Tracking (MHT) [202] that identify and track individual targets. For example, Gruteser and Hoh [117] use the MHT algorithm to process the anonymous periodic positional Global Positioning System (GPS) information generated by a small number of students on a University campus. Despite the relatively poor positional accuracy of GPS, most of the anonymous samples could be assigned to the correct source and the behaviour of individual students over the experiment duration could be determined. Wiedersheim et al. [259] apply MHT to simulated vehicular networks and evaluate the tracking capabilities of an adversary across several thousand different simulations. The authors found that for all anonymous beaconing rates in excess of one message per second, the average simulated

vehicle was tracked with an 80% success rate.

At close range, the adversarial model and the user model for V2X communication is indistinguishable. In both cases the subject is attempting to determine the precise behaviour of nearby road users and will succeed in this task because it is the intended functionality and purpose of V2X communication. Similarly, a global adversary with uninterrupted access to the V2X messages sent by all road users will always succeed in identifying and tracking individual users. The scope of achievable inter-vehicle privacy in V2X is therefore limited to the intermediate adversary that has only a temporally intermittent, or spatially partial observational capacity.

Concerning the scope of achievable privacy with regards to privileged V2X adversaries that reside within the infrastructure, there is also a conflict between the functional system requirements and user privacy. In particular, to ensure that CAM can be trusted to determine safety-critical vehicle behaviour, V2X architectures require Sybil attack resistance, the removal of misbehaving users and optionally, law enforcement support. These requirements preclude any truly anonymous scheme and require, at the least, a mechanism for revoking user credentials based on some messages that can be linked to a specific source [244]. In addition, any mechanism for user revocation or exclusion must be sufficiently efficient that the requirements for high message throughput and limited, intermittent vehicle connectivity can be met.

Across much of the literature [23, 102, 189, 84] and in both of the leading ITS standards [86, 29] the main technique for achieving privacy in V2X communication is pseudonym certificates. Pseudonyms allow vehicles to send messages without revealing their identity, whilst still remaining accountable.

Pseudonyms

The European Secure Vehicle COMmunication (SeVeCOM) project [158] aimed to address the security and privacy of future vehicular communication networks. In particular, SeVe-

COM performed an extensive review of the potential vulnerabilities in ITS, developed corresponding threat models and then proposed a number of appropriate security mechanisms based on the prior art [244, 159]. Borrowing from the wider literature on privacy and trust [48], SeVeCOM established the mainstream use of pseudonyms for privacy protection in ITS.

Pseudonyms provide privacy mechanisms for V2X in two dimensions. Firstly, in both the leading European [86] and U.S. [29] standards for ITS security architecture, and across much of the literature [23, 188, 102, 208, 256, 189], pseudonyms are used to prevent any single authority from linking vehicle messages to the corresponding vehicle identity. The role-separation of certificate authorities typically provides vehicle privacy with the following two-stage issuance process. First of all, the long-term enrolment credentials issued to road users by the enrolment authority are pseudonymous and do not indicate the canonical identity of the vehicle. Independently, one or more authorisation authorities provide pseudonymous authorisation certificates for specific services to any authorised user who also presents a valid enrolment credential.

The primary use of pseudonyms in V2X is to provide privacy for inter-vehicle communication. Generally, the safety-critical broadcast messages that are sent by vehicles are signed using pseudonymous authorisation certificates. To prevent long-term tracking, vehicles implement a pseudonym change strategy that aims to hide their overall behaviour. The idea is that if a vehicle changes pseudonym during a period when it is not being observed, then the adversary will be unable to precisely determine whether multiple pseudonyms belong to the same source. Because of the importance of preventing long-term vehicle tracking in V2X, considerable attention has been given to developing pseudonym change strategies. A recent pre-standardisation survey on pseudonym change methods by ETSI [84] has identified 7 different categories in the literature:

Fixed Parameters The simplest approach to pseudonym change is to define a fixed

change parameter. Often the parameter is time-based such as one pseudonym every 5 minutes [29], but using a fixed number of messages or a measure of distance has also been considered. To reduce the potentially large storage and bandwidth overhead associated with short-lived pseudonym certificates, Eckhoff et al. [77] propose a time-slotted certificate pooling approach that provisions small sets of time-limited certificates with multiple, distinct validity periods. Pseudonyms are reused at multiple periods during the pool duration to allow for fewer overall certificates. The U.S. DOT preferred approach for pseudonym change [29] is also based on time-slotted certificate pooling, every vehicle is given a set of 20 pseudonyms per week and pseudonyms are changed every 5 minutes. Limited simulation results by Wiedersheim et al. [259] indicate that for fixed pseudonym change intervals greater than 4 seconds, vehicles can be tracked with almost 100% success.

Random Parameters To disrupt the predictability of the fixed parameter change strategy, each pseudonym change point may also include a random quantity. Pan et al. [186] show that this approach provides a larger anonymity set for each vehicle, across a range of different simulations, than using a fixed certificate period.

Silent Periods During silent periods vehicles do not send any broadcast messages, but they still receive and process them. Huang et al. [160] were the first to propose a random silent period between pseudonym changes and showed that an adversary was significantly less able to track vehicles over multiple pseudonym periods. The Society of Automotive Engineers (SAE) BSM message set standard [207] specifies a silent period of between 0 and 13 seconds between pseudonym changes. The main drawback to silent periods is that they involve voluntarily withholding information that may prevent an accident. Lefévre et al. [157] consider an adaptive pseudonym change strategy that considers the current ability of the vehicle to operate safely before changing pseudonym with a silent period. The authors simulate their strategy

and show that the safety is comparable to using no silent period, whilst the privacy is nearly as strong as the SAE approach.

Vehicle Centric In this technique, vehicles independently choose when and where to change pseudonym based on their local circumstances. Li et al. [161] propose a strategy called swing in which vehicles only change pseudonym when changing speed and direction. In combination with silent periods, simulations indicate that swing provides a higher entropy anonymity set than changing pseudonyms randomly.

Density Based In the density based pseudonym change strategy [187], vehicles only change identity when there is a sufficiently large number of neighbouring vehicles.

Mix Zones Beresford and Stajano [15] were the first to introduce mix zones as a general technique for location privacy in pervasive computing. A mix zone is a spatial region in which users do not provide any location information. Provided that vehicles change to a new pseudonym whenever they enter a mix zone, an adversary cannot distinguish the vehicle from any other who was in the zone at the same time. This also means that the adversary is unable to link vehicles going into the mix zone with those coming out of it. Buttyán et al. [37] evaluate vehicular mix zones as a strategy for pseudonym change and show that the resulting privacy is highly variable depending on the traffic flow and the number of antennas that are owned by the adversary.

Pseudonym Swap Li et al. [161] propose a pseudonym change strategy called swap in which neighbouring vehicles that simultaneously change pseudonyms have a 50% probability of exchanging identity. The author's simulations indicate that swap outperforms both vehicle-centric and random pseudonym change strategies.

Despite the considerable interest in pseudonym change strategies for providing location privacy in V2X, deciding the optimal strategy remains an open problem. Without a standardised test, game or simulation scenario, there is no common platform against

which to benchmark different approaches. Ma et al. [167] develop a generic quantitative location privacy metric for V2X communication that is based on measuring the uncertainty of linking individual vehicles with complete journeys. Freudiger et al. [101] develop a game-theoretic model of pseudonym change in which they consider selfish, independent mobile nodes that decide locally on whether to change pseudonym or not. The authors use the results of their model to improve the swing protocol of Li et al. [161] so that it also considers optimal location privacy for non-cooperative vehicles.

3.2.3 Security Architectures

One of the first architectures to explicitly consider the unique security challenges of vehicular networks was by Zarki et al [268]. The authors draft a scheme in which vehicles use digital signatures for authentication and a PKI is used to manage trust. Independently, Gollan and Meinel [113] also propose digital signatures for vehicles and briefly discuss the privacy implications. Soon afterwards, the European SeVeCOM project undertook the development of a complete security architecture for V2X [213] that was based on an extensive requirements analysis process and that harmonised a number of prior techniques.

SeVeCOM

SeVeCOM identified a set design principles that have widely influenced the development of standards and literature. In particular, SeVeCOM called for the separation of privilege, and of the roles of authorities and infrastructure, providing increased security, privacy and fault tolerance. SeVeCOM also called for privacy that is at least the same level that is currently afforded without vehicular networks. Vehicles must be accountable for the messages they send, but they must also be appropriately protected against individual, commercial and government organisations that would seek to arbitrarily track them.

The SeVeCOM architecture [213] is as follows. All users, vehicles and infrastructure nodes have a unique identifier and a public key pair. There is a simple hierarchical PKI

in which groups of road users all trust and are registered to a particular authority. The presence of an online authority is not required and vehicle private keys are protected by a tamper-resistant secure hardware element that includes a cryptographic processor. Broadcast messages are signed to ensure the authentication and authorisation of the sender, as well as to provide integrity assurance. SeVeCOM recognised that the traditional CRL approach for revocation does not scale well to the extreme size and mobility of vehicular networks and proposed a number of alternative approaches [197].

To identify the most suitable authentication techniques for vehicular communication, SeVeCOM considered the performance of three different digital signature schemes. The RSA, DSA and ECDSA signature schemes were evaluated with respect to their signature and public key sizes, the time taken to generate and verify signatures and their scalability [213]. Across all measures, ECDSA outperforms DSA. In contrast to RSA, ECDSA has slightly slower verification, but signature generation is an order of magnitude faster, the public key and signatures sizes are more than 3 times smaller and the algorithm scales much more favourably to larger key lengths. Overall, SeVeCOM determined that ECDSA is superior to either DSA or RSA for V2X communication and incorporated the algorithm into its reference security architecture.

ETSI Standard

Today, many of the SeVeCOM design principles, security mechanisms and reference architecture components are used by the leading security architecture standards for ITS. Both the European ETSI TS 102 731 standard [85], and the latest U.S. DOT approach based on the Security Credential Management System (SCMS) [258, 29], are based on ECDSA digital signatures and a hierarchical PKI that facilitates and manages the certificates that are necessary for establishing trust between road users. Both standards use role separation for privacy, and allow for the selective depseudonymisation of misbehaving users.

The ETSI security architecture is based on the simple PKI proposed by the SeVeCOM

project in [213]. In the ETSI PKI trust model, there is one or more root certificate authorities (RCA), enrolment authorities (EA) and authorisation authorities (AA). The RCA defines a common policy and signs the certificates of the EA and the AA. The EA maintains the canonical identity of registered vehicles and correspondingly issues and manages long-term enrolment credentials. The AA provides short-lived anonymous pseudonym certificates to any authorised vehicle that also presents a valid long term enrolment credential. The ETSI-preferred approach to revocation [87] is passive and based on the issuance of short-lived pseudonym certificates [198] that are not replenished for vehicles that have misbehaved. A standard for pseudonym change has not yet been finalised, but the start-of-the-art has been extensively surveyed [84] and a number of recommendations made. In particular, the need to set a maximum pseudonym duration and the importance of minimising the number of simultaneously valid pseudonyms has been noted.

Security Credential Management System

The leading security architecture candidate for V2X communication in the U.S. is the SCMS [29] developed under a cooperative agreement with the U.S. DOT. In contrast to the ETSI approach [85], SCMS has a different PKI model, certificate scheme and revocation mechanism. In addition to the root, enrolment and authorisation authorities, SCMS introduces a number of additional roles. Specifically, the SCMS PKI has 13 additional roles that include one or more Pseudonym CA (PCA), Linkage Authority (LA) and Misbehavior Authority (MA).

As-per the SeVeCOM reference architecture [213], vehicles are provisioned with a long-term certificate by the enrolment authority and request short-lived pseudonym certificates from the authorisation authority, termed the Registration Authority (RA) in SCMS. The RA does not provide vehicles with pseudonyms directly but instead forwards requests for certificates to the PCA.

At the heart of SCMS is a unique "butterfly-key expansion" mechanism that allows vehicles to request an arbitrary number of pseudonyms, with only a single request, such that no single authority can link multiple pseudonyms to a single source. The vehicle provides the RA with a "caterpillar" base point $A = a \times G$ and an expansion function $f_k(l)$, that is a pseudo-random permutation in the integers mod l. The RA uses the point A, and the function $f_k(l)$, to generate "cocoon" public keys such that each pseudonym B is computed $B = A + fk(l) \times G$ and the corresponding private key b is computed by the vehicle as $b = a + fk(l)$. The RA sends the cocoon public keys, individually and in a random order, to the PCA who similarly exploits modular addition to derive final pseudonym public key values. The pseudonym public keys and their derivation factors are encrypted before being returned to the RA for distribution to the requesting vehicle.

Unlike the SeVeCOM reference architecture [213], SCMS does use conventional CRL for revocation. However, to alleviate some of the scalability issues associated with having multiple pseudonym certificates per vehicle, SCMS introduces a novel linkage-value certificate mechanism that is used to reduce the CRL size. For each pseudonym certificate provided to a vehicle, SCMS inserts a unique linkage value that can be used to revoke all of the pseudonyms that have a validity equal to or later than some time. Linkage values are created by the PCA according to pre-linkage values that are generated for each vehicle by the pair of linkage authorities, LA1 and LA2. For revocation, both LAs, the PCA and the RA must collaborate with the misbehaviour authority MA who publishes pre-linkage value seeds on the CRL, such that all future pseudonym certificates on the corresponding vehicle are revoked.

For pseudonym change, SCMS recommends the following adaptation of the C2C-CC model [84]. Each week, a batch of 20 or more pseudonym certificates are simultaneously valid and vehicles are free to switch between the pseudonyms within the batch using any strategy. Vehicles may be preloaded with a total of 1-3 years worth of pseudonym batches.

Beyond the Standards

The main limitations of the leading V2X standards [85, 29] are revocation, pseudonym resolution and privacy against curious or malicious certificate authorities. Petit et al. [189] survey the wider literature on cryptographic schemes for vehicular communication and identify a number of alternative techniques.

One method for improving privacy is to decentralise the roles of certificate generation and revocation to the vehicle nodes. All of these approaches are based on the assumption of a tamper-proof device onboard each vehicle and typically rely on the vehicle correctly forwarding revocation messages to the secure device. In critique of the limitations of CRL and short-lived pseudonym certificates, Raya et al. [197] propose a decentralised revocation protocol. The certificate authority must know the long-term identity of the misbehaving vehicle and then uses roadside infrastructure to send an encrypted revocation message. Vehicles that receive a revocation message are individually responsible for deleting their key material so that they are no longer able to broadcast authenticated messages.

Similarly, but without the need for pseudonym resolution, Förster et al. [98] propose the Revocation Without Resolution (REWIRE) protocol in which revocation demands are sent based on unresolved pseudonyms. To combat non-compliant vehicles, the authors suggest using dummy keep-alive messages that are indistinguishable from a revocation request until they are processed by the trusted hardware. Whitefield et al. [257] use symbolic formal verification methods to analyse the REWIRE revocation protocol and identify a number of security flaws. The authors develop the improved Obscure Token (O-Token) protocol that offers a strong, verifiable guarantee of vehicular revocation without exposing the long-term identity. Zeng [269] present the Pseudonymous PKI (PPKI) architecture which is a general pairing-based solution for ubiquitous computing that decentralises pseudonym generation whilst retaining centralised tracing and revocation capabilities. Armknecht et al. [8] apply PPKI to vehicular communication and propose a related security architecture.

Uniquely, revocation does not require cooperation from the misbehaving vehicle but is based on re-keying the uncompromised nodes with updated system parameters.

A number of techniques for providing enhanced privacy against dishonest or colluding certificate authorities have also been developed. Schaub et al. [208] propose a scheme that prevents individual issuing authorities from learning linking information, but that also permits conditional depseudonymisation based on information that is embedded directly into pseudonym certificates. The scheme is based on so-called V-tokens that comprise a long-term vehicle identity and a randomisation factor, all encrypted with a multiparty revocation public key. V-tokens are blindly verified and signed by the certificate authority, and then signed V-tokens are used to anonymously request pseudonyms from the pseudonym authority. To resolve the identity of a misbehaving vehicle, multiple revocation authorities must collaborate to decrypt the V-token that is embedded into the corresponding pseudonym certificate.

PUCA [102] has stronger privacy guarantees than Schaub et al's V-token scheme and protects users even against colluding certificate authorities. Vehicles use fully anonymous credentials for authentication with the certification infrastructure, yet retain efficient standards-compliant signatures for inter-vehicle communication. Revocation is based on the REWIRE protocol and depends on vehicles that have tamper-proof onboard hardware and will delete their key material when requested. Also based on anonymous credentials, Whitefield et al. [256] use Direct Anonymous Attestation (DAA) to fully decentralise the management of pseudonym certificates such that vehicles are entirely responsible for the generation, anonymity, linkability and revocation of their own pseudonyms. This approach is critically dependent on the integrity of tamper-proof hardware onboard each vehicle and demands that revocation messages are reliably forwarded to the trusted hardware. Chen et al. [50] also use DAA and propose a scheme that, whilst retaining centralised revocation and optional depseudonymisation, provides a mechanism for detecting vehicles

that abuse their anonymity to send multiple messages relating to the same event.

Chapter 4

Dismantling the AUT64 Automotive Cipher

In this chapter we present a thorough analysis of a popular and previously unstudied vehicle immobiliser system. We reverse engineer the firmware recovered from a vehicle immobiliser unit and expose the full details of the proprietary automotive block cipher, AUT64. Unlike some of the other automotive ciphers that have already received attention in the literature [28, 134, 241], AUT64 has a large 120 bit secret key that is not susceptible to exhaustive search. We evaluate the security of our target immobiliser system and identify a number of implementational, key management and cryptographic weaknesses. We combine the results of our evaluation to present several practical key recovery attacks that, in the worst-case for an attacker, recover the full 120 bit secret key in just $2^{48.3}$ encryptions. Our results contribute to mounting evidence that obscurity is not a substitute for standardised algorithms and peer-reviewed protocols. Like many others, we urge the automotive industry to discontinue the use of proprietary cryptographic implementations and to avoid simple key management blunders. Our results also make a more general contribution to the literature on the cryptanalysis of generalised Feistel ciphers with key-dependent permutations and S-Boxes.

4.1 Motivation

Since the study of the DST immobiliser system by Bono et al. [28] first exposed the practice of using weak, proprietary cryptography for automotive applications, the academic community has duly proceeded to evaluate a number of other high-profile systems [105, 249, 248, 104, 46]. Such work ensures that the systems we depend upon to secure our vehicles, workplaces and critical national infrastructure [104] offer a degree of security that is proportional to the value that they protect. Whilst research indicates [247] that the majority of vehicles sold in Europe between 1995 and 2015 are fitted with an immobiliser system based on either the Hitag2 [248] or Megamos Crypto [249] encryption algorithm, there are still a number of relatively widespread proprietary algorithms and automotive systems that have received less attention. In particular, the AUT64 algorithm was first identified by Garcia et al. [105] in their case study of vehicle RKE systems. AUT64 is used in millions of VW group vehicles sold between 2004 and 2006 but, owing to the use of a single global master key for all vehicles with this system, the algorithm was not evaluated at this time. In this work we discover a vehicle immobiliser system based on the same algorithm. In the absence of global master key for this application, we are motivated to recover the full details of the AUT64 algorithm and to thoroughly evaluate the security of the system.

4.2 Contributions

We make three main contributions in this chapter. First, in Section 4.4 we present the results of reverse engineering the AUT64 algorithm from a Mazda immobiliser system. We reveal complete details of the AUT64 block cipher and its associated authentication protocol. Secondly, we present a thorough evaluation of the immobiliser system that includes an extensive cryptanalysis of the AUT64 algorithm in Section 4.5 and Section 4.6.

We are able to demonstrate a number of concrete weaknesses in the algorithm. The AUT64 Feistel network function comprises a compression function and a substitution-permutation network that have the following weaknesses:

- The input to the compression function in the first round can be precisely controlled by a chosen-plaintext adversary.
- The output of the compression function is not uniformly random.
- The output of the substitution-permutation network is cryptographically weak when its input is nibble-wise symmetric.

In addition, regarding the overall design of the AUT64 block cipher we show that:

- By exploiting the weaknesses we identified in the Feistel network function, we can bias the output of the early rounds of encryption.
- The cryptographically weak output from an early round of encryption can be identified by statistically analysing small sets of ciphertexts.
- The cipher has certain weak keys.

In Section 4.6.4 and Section 4.6.5, we develop a novel attack based on integral cryptanalysis [151] that allows us to determine several elements of the secret key. We show that 8 round AUT64 has weaknesses that reduce the entropy of the secret key from 120 bits to no more than 57.5 bits in a chosen-plaintext setting. These results are of independent interest with respect to the cryptanalysis of Feistel network ciphers with key-dependent permutations and S-Boxes.

Thirdly, in Section 4.7.2 we evaluate the overall security of the target Mazda immobiliser system and identify the following implementational and key management weaknesses:

- The key management scheme is flawed and significantly reduces the security of the system. The entropy of the permutation key part is reduced from $8! \approx 2^{15.3}$ to just

16 possible key values per automotive manufacturer, with one key being used per vehicle family.

- An additional 32 bits of the secret key are derived from the public transponder ID using a manufacturer-wide key derivation function.
- There are indications that a large part of the key is constant across different vehicles by the same manufacturer.

Based on the cryptographic and implementational weaknesses of AUT64, which we summarise in Table 4.1, we present two practical attacks. Our first attack targets the full 24 round AUT64 implementation as found in the studied immobiliser system. Our attack requires 2 authentication traces and $2^{48.3}$ encryptions to recover the 120-bit secret key in the worst case. In practice, the security is likely much worse owing to bad key management. Our second attack exploits knowledge of the 32 bits of secret key that can be derived from the public transponder ID to break 8 rounds of AUT64 using just 512 chosen plaintexts and, in the worst case, $2^{37.3}$ offline encryptions. In most cases, the attack can be executed within milliseconds on a standard laptop.

Weakness	**Section**	**Keyspace (bits)**
None	4.4	120
High-level Analysis	4.4.2	91.5
Permutation Key Size	4.5.1	88.5
Compression Function Weakness	4.5.2	$\leqslant 87.5$
Permutation Weakness	4.6.1	85.7
Compression Function Weaknesses	4.6.2, 4.6.3	$\leqslant 57.6$
Integral Cryptanalysis	4.6.4, 4.6.5	$\leqslant 78.8$
Key Derivation Weakness	4.7.1	59.5

Table 4.1: A summary of the cryptographic AUT64 weaknesses identified in this work and the corresponding impact on the security of the cipher.

4.3 Notation

In this chapter we make use of the following notation conventions. We let $\wedge$, $\vee$, $\oplus$, $\ll$ and $\gg$ define the bitwise AND, OR, XOR, left-shift and right-shift operations, respectively. We use $\|$ to denote bitwise concatenation and we use the term *symmetric byte* to refer to a byte $b \in \mathbb{F}_2^8$ of the form $n \,\|\, n$ where $n \in \mathbb{F}_2^4$. In the context of byte strings or lookup tables, we specify an index using square brackets. We define two functions $\texttt{msb}_m$ and $\texttt{lsb}_m$ that return the m most and least significant bits, respectively. When the operand is a single byte we also define the functions `un` and `ln` to mean $\texttt{msb}_4$ and $\texttt{lsb}_4$, respectively. Finally, where $\mathcal{M}$ is some set we use $m \xleftarrow{\$} \mathcal{M}$ to denote the assignment to m an element in $\mathcal{M}$ chosen uniformly at random.

When describing the AUT64 block cipher, we use the taxonomy and language of generalized Feistel networks developed by Schneier and Kelsey [211].

4.4 AUT64

In this section we present the complete details of the AUT64 block cipher and its associated authentication protocol. AUT64 is a 64-bit block cipher with a 120-bit secret key that was first identified by Garcia et al [105] as the algorithm used to secure most VW Group RKE systems manufactured between approximately 2004 and 2006. The details of AUT64 were not studied further at this time owing to the use of a single, global master key. As a proprietary algorithm, the only prior publicly available information on AUT64 are from the patent application [34] and the product datasheet [224].

In this work, we present the AUT64 algorithm as implemented by the Atmel TK5561 [199], an automotive transponder package for the Atmel e5561 cryptographic IDentification IC (IDIC) [224]. The TK5561 transponder is used in a number of Mazda, Ford and Proton vehicle keys, which are shown in Figure 4.2. The e5561 IDIC combines the AUT64

block cipher and a custom made challenge-response protocol to provide an authentication mechanism that is intended for use in "high-security" automotive applications such as vehicle immobilisation.

Concretely, AUT64 is a 64-bit Unbalanced Feistel Network (UFN) block cipher with a 120 bit key length. In the TK5561 transponder package, the cipher is configurable to use either 8 or 24 rounds. AUT64 has also been found using a 12 round implementation in the VW Group RKE system where the cipher was first discovered. The AUT64 key space is the triplet of all 32-bit binary strings, all 8 element permutations and all 16 element permutations $\mathcal{K} = \langle \mathbb{F}_2^{32}, P_8, P_{16} \rangle$. In this work we refer to these key parts as the compression function key, permutation key and substitution key, respectively. The 120 bit key size is the sum of the 32, 24 and 64 bits occupied by the $\mathbb{F}_2^{32}$, P_8 and P_{16} key parts respectively.

Make	**Models**	**Years**
Mazda	323	1999-2003
	626	1999-2002
	Demio	1999-2001
	Miata/MX-5	2000-2005
	MPV	2000-2006
	Premacy	2000-2004
	121	1999-2001
	BT-50	2006
Ford	Ranger	2006
Proton	415	1998
	416	1998

Table 4.2: Vehicles known to use TK5561 transponder keys.

4.4.1 Reverse Engineering an Immobiliser System

To recover the AUT64 algorithm we reverse engineered the immobiliser unit firmware from a vehicle that uses the TK5561 transponder. Specifically, we recovered the firmware from a Mazda "Module 142" immobiliser system using a standard programmer. Since this immobiliser system was built using an off-the-shelf Motorola MC68HC05B6 microcontroller, we were able to use the IDA Pro disassembler to analyse the firmware image.

By disassembling the firmware image we were able to fully reconstruct the AUT64 algorithm and the challenge-response protocol used by the immobiliser system. We found

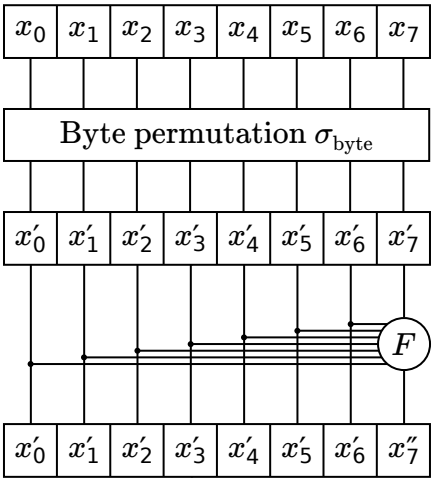
that the Mazda system uses a 24 round profile of the AUT64 encryption algorithm and challenge-response authentication protocol specified by the Atmel TK5561 transponder package. We proceeded to create a reference implementation of AUT64 in both Python and C which we then used to develop, test and evaluate our attacks. We verified the results of our firmware analysis against the available TK5561 transponder documentation [34, 224] and an implementation that was developed for working with VW Group RKE systems [105]. Our analysis of the firmware also allowed us to locate the permutation and S-Box key parts from our target system:

- The permutation key part k_σ, also located in page 6 in the transponder's memory, is found in the first 32 bits of the microcontroller's Programmable Read-Only Memory (PROM) starting at the address `0x0800`. The k_σ we recovered, `0x25763041`, is a cyclic permutation [24] with no fixed points.
- The substitution key part k_τ, also found on pages 7 and 8 in the transponder's memory, is located in the 64 bits of PROM that immediately follow the permutation key part. The k_τ we recovered, `0x436aef12d5890c7b`, is bijective, as expected, but otherwise unremarkable.

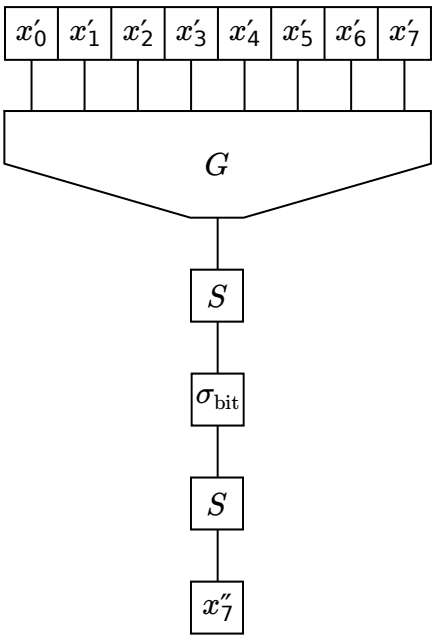
Surprisingly we did not find the 32-bit compression function key part k_G stored in the immobiliser system firmware. Instead, we discovered that k_G is computed from the public transponder ID that is transmitted each time the transponder initiates the TK5561 challenge-response protocol. k_G is computed using a proprietary, manufacturer-wide key derivation function that we detail further in Section 4.7.2.

4.4.2 The AUT64 Block Cipher

In this section we fully specify the AUT64 block cipher. We begin with a description of the high-level Feistel network and a secret-key specification and then we describe the full key-dependent structure.



(a) The high-level AUT64 Feistel network structure. In each round, all bytes in the state are permuted according to the byte permutation σ_{byte} and then the seventh byte of the permuted state x'_7 is encrypted using the Feistel function F.



(b) The AUT64 Feistel function F.

Figure 4.1: The AUT64 Feistel network construction (a) and Feistel function F (b).

The AUT64 cipher state and secret key structure are defined as follows.

Definition 4.4.1 (AUT64 Cipher State) We define an AUT64 state X as an element in $\mathbb{F}_2^{64}$ where X is composed of eight bytes $x_0, \ldots, x_7$, each an element in $\mathbb{F}_2^8$.

Definition 4.4.2 (AUT64 Secret Key) We define an AUT64 secret key as a triplet $K \in \langle k_G, k_\sigma, k_\tau \rangle$ where:

- The compression function key part $k_G \in \mathbb{F}_2^{32}$ is a bit string that is used to by the key schedule to create round keys.
- k_σ is an 8 element permutation that defines both the Feistel network byte permutation σ_{byte} and the Feistel function bit permutation σ_{bit}.
- k_τ is a 16 element permutation that defines the 4×4 S-Box τ that is repeatedly applied in the Feistel function F.

With respect to Definition 4.4.1 and the block cipher taxonomy we introduced in

Section 2.3.2, AUT64 is a UFN block cipher where the source block is the 64-bit cipher state $X = (x_0 \dots x_7)$ and the target block is the seventh byte of the permuted state $X'[7] = x'_7$. As shown in Figure 4.1a, the high-level Feistel network structure of AUT64 comprises two operations:

1. The byte permutation $\sigma(X)$ which takes as input the cipher state X and applies the key-dependent byte permutation σ_{byte}. The output is the permuted state $X' = (x'_0 \dots x'_7)$.
2. The Feistel function F that takes as input the permuted state X' and produces one encrypted output byte x''_7.

In each round, the cipher state is first permuted $\sigma_{\text{byte}} : (x_0, \dots, x_7) \rightarrow (x'_0, \dots, x'_7)$ and then the Feistel function $F : (x_0, \dots, x_7) \rightarrow x''_7$ is applied. The UFN construction of AUT64 necessitates that, at a minimum, for each input bit to be replaced with encrypted output from the Feistel function F the cipher must be run for 8 rounds. In more detail, we can determine that the byte permutation σ_{byte} should have a cycle length of 8 and therefore be free of fixed points. If σ_{byte} had any fixed points then whole bytes of the plaintext would always appear in the ciphertext, regardless of how many rounds were applied. In the remainder of this work we assume that the AUT64 permutation key part k_σ always specifies a cyclic permutation of length 8 with no fixed points.

With respect to Definition 4.4.2, the AUT64 patent application [34] specifies a key generation and diversification process that relates to the structure of an AUT64 key. The AUT64 key generation procedure is as follows:

- The so-called "random key", or compression function key part k_G, is a random bit string that is generated from the DES encryption of a seed that is to be randomly chosen by the automotive manufacturer.
- The"family key", corresponding to the permutation key part k_σ, is partially allocated by Atmel and partially chosen by the automotive manufacturer. Specifically, each

manufacturer is allocated 16 different family keys and is expected to use a different key per vehicle family.

- The "user key", which corresponds to the substitution key part k_τ, is generated using a proprietary "special method" that is not otherwise specified. The patent claims that the user key will only repeat after 20.9×10^{12} keys have been generated. This claim would suggest that the entire 4×4 bijective S-Box space is utilised when choosing the user key.

Definition 4.4.3 (AUT64 Feistel Function) As shown in Figure 4.1b, the AUT64 Feistel function comprises the following four operations:

1. The compression function G that takes as input the permuted cipher state $(x'_0, \ldots, x'_7)$ and outputs a compressed byte g.
2. The substitution operation $S = \tau\Big(\texttt{un}(g)\Big) \parallel \tau\Big(\texttt{ln}(g)\Big)$ which is composed from the nibble-wise application of the key-dependent 4×4 S-Box τ to the upper and lower nibble of the compressed byte g.
3. The permutation operation σ_{bit} that takes as input the substituted, compressed byte $S(g)$ and applies the bitwise permutation σ_{bit} that is defined by the permutation key part k_σ.
4. The substitution operation is applied again but to the output from the bitwise permutation. This final operation produces the encrypted byte that is output by the Feistel function: $x''_7 = S\left(\sigma_{\text{bit}}\Big(S(g)\Big)\right)$.

As shown in Figure 4.2, the AUT64 compression function G computes two nibble-wise XOR sums of values chosen from the key independent lookup table T_{offset}. Specifically, for each input byte $x'_0, \ldots, x'_7$, the function G computes an upper and lower-nibble round key. Each 4-bit round key is used alongside the corresponding upper or lower input byte nibble to select a value from the table T_{offset}. Each value selected from T_{offset} is added, modulo 2, into the corresponding upper or lower output register.

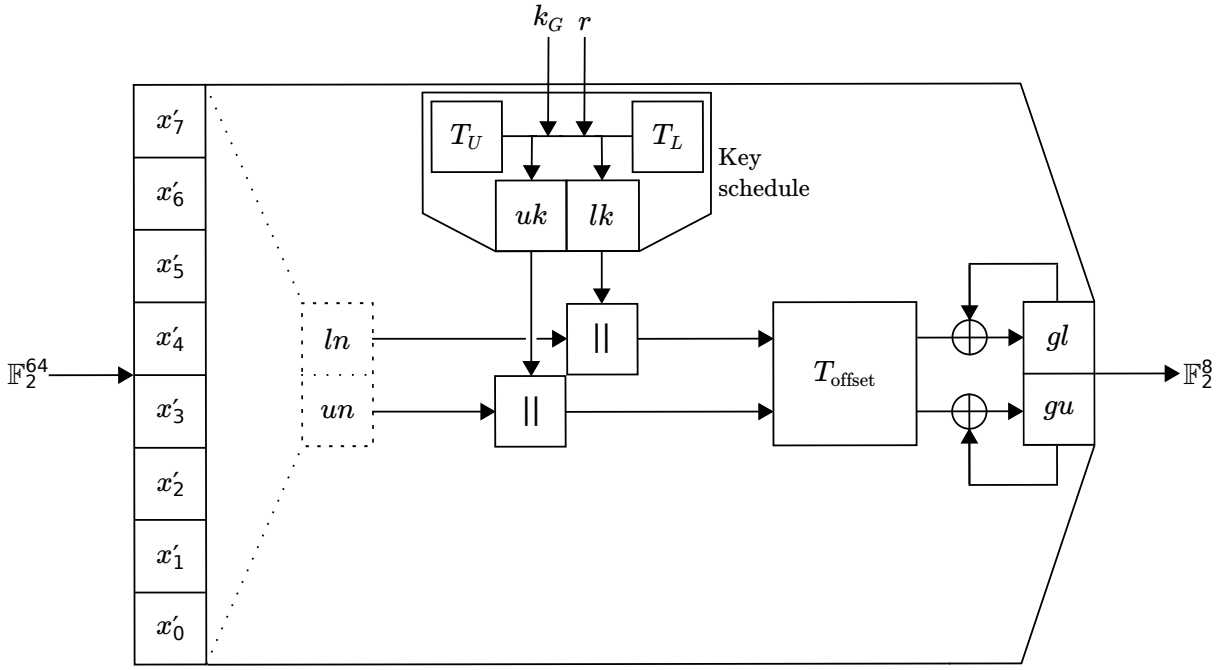


Figure 4.2: The AUT64 compression function G that includes a key scheduling mechanism. The dotted lines indicate that each input byte $x'_0, \ldots, x'_7$ is individually input to the remainder of the function.

The compression function lookup table T_{offset} is a 16×16 array of nibble values that is symmetric about its descending diagonal axis. When accessing the table, the compression function G uses the round-key nibble to select a row and the input-byte nibble to select a column. Although not used in this vehicle immobiliser application, T_{offset} indicates that AUT64 decryption is possible as the output nibble from T_{offset} and the round-key nibble can be used to uniquely determine the input-byte nibble that was encrypted. We specify the complete compression function lookup table that we recovered from the Mazda immobiliser system firmware in Appendix A.

We now specify the AUT64 key scheduling algorithm. The AUT64 key schedule is part of the compression function G that is shown in Figure 4.2. In each round, the compression function derives a unique round key from the key part k_G and the key scheduling tables T_U and T_L. For every round of AUT64 encryption, each table T_U and T_L specifies a unique

permutation of the compression function key part k_G. Two round keys are computed for each round as T_U and T_L are used to compute the round key that is used to encrypt the upper and lower nibble in each input byte, respectively. The full key scheduling tables from the Mazda immobiliser system firmware we analysed can be found in Appendix A.

Definition 4.4.4 (AUT64 Round Keys) Let r be the round number and i the input byte index, then the upper and lower round key nibbles, respectively, are computed as follows:

$$uk(k_G, r, i) = k_G\Big[T_U\Big[(r \times 8) + i\Big]\Big] \qquad lk(k_G, r, i) = k_G\Big[T_L\Big[(r \times 8) + i\Big]\Big]$$

Definition 4.4.5 (AUT64 Compression Function) The AUT64 compression function G takes as input the permuted state $x'_0, \ldots, x'_7$, the key part k_G, the round number r and the key-independent lookup tables T_{offset}, T_U and T_L. Eventually, G outputs a single byte that is the concatenation of two 4-bit variables, gl and gu, that are computed as follows:

$$gu = \bigoplus_{i=0}^{7} T_{\text{offset}}\Big[uk(k_G, r, i) \parallel un(X'_i)\Big] \qquad gl = \bigoplus_{i=0}^{7} T_{\text{offset}}\Big[lk(k_G, r, i) \parallel ln(X'_i)\Big]$$

Ideally, the compression function G would output a uniformly random 4-bit value. It is notable however that as the first row and column in T_{offset} contain only the value zero, this is not the case. Indeed, if either the round-key or input-byte nibble has the value zero then for that particular input G will always add zero to the respective output register, leaving it unchanged. We exploit this weakness further in Section 4.5.2.

4.4.3 Authentication Protocol

This section describes the challenge-response authentication protocol that takes place between the vehicle immobiliser unit and the TK5561 transponder. According to the

patent, the objectives of the protocol are to provide a method for authentication and to prevent known and chosen-plaintext cryptographic attacks [34].

Definition 4.4.6 (AUT64 Authentication) We define AUT64 authentication as the quartet of algorithms $\mathbb{A} = \Big(\mathrm{Enc}_{\mathrm{AUT64}}, C, R, H\Big)$, where for all AUT64 keys $K \in \langle k_G, k_\sigma, k_\tau \rangle$, for all checksums $h \in \mathbb{F}_2^5$ and for all nonces and challenges $(X, Y) \in \mathbb{F}_2^{64}$:

- $\mathrm{Enc}_{\mathrm{AUT64}} : X \mapsto \mathrm{Enc}_{\mathrm{AUT64}}(K, X)$ is the AUT64 encryption algorithm.
- $C_{k_G} : Y \mapsto C(k_G, X)$ is the protocol challenge algorithm which is keyed with the compression function key part $k_G \in K$.
- $R_{k_G} : X \mapsto R(k_G, Y)$ is the protocol nonce-recovery algorithm which is keyed with k_G and computes the preimage of C_{k_G} such that $\forall k_G \in \mathbb{F}_2^{32}$ and $\forall X \in \mathbb{F}_2^{64}$, the following consistency equation holds $X = R_{k_G}\Big(C_{k_G}(X)\Big)$.
- $H : h \mapsto H(Y)$ computes the hamming weight of Y.

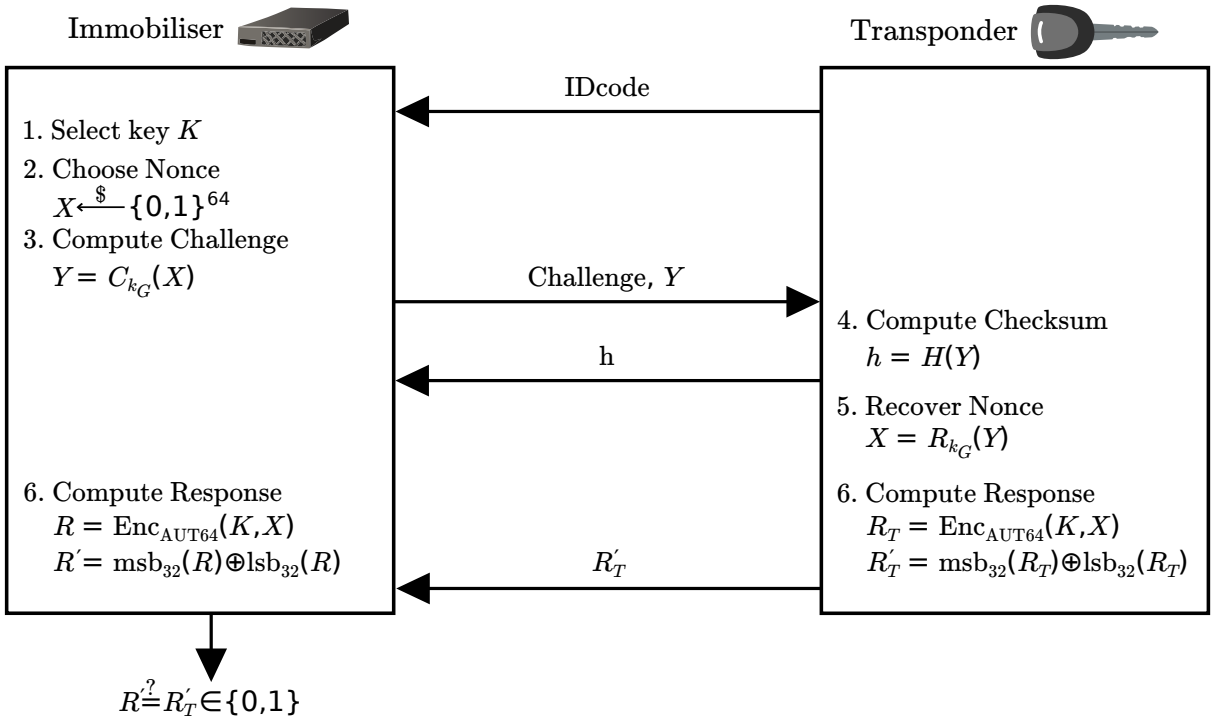


Figure 4.3: The TK5561 authentication protocol.

During installation of the vehicle immobiliser system, each authorised transponder key has its IDcode and AUT64 key K stored within the immobiliser unit. As we discuss in Section 2.1.1, during normal operation the transponder is energised by the immobiliser system antenna when the vehicle key is placed in the ignition barrel. The TK5561 immobiliser authentication protocol, shown in Figure 4.3, comprises the following six steps:

1. Once energised, the transponder initiates the protocol by sending its unique IDcode value. When an authorised transponder sends its IDcode, the immobiliser unit looks up the corresponding AUT64 key K. To account for spare keys, each immobiliser unit may have several different authorised transponder IDcodes.
2. The immobiliser unit chooses a random 64-bit nonce X.
3. The immobiliser computes the challenge $Y = C_{k_G}(X)$, by applying the challenge algorithm to the nonce X, and then sends Y to the transponder.
4. The transponder computes the hamming weight $h = H(Y)$ of the challenge Y and then sends the result to the vehicle immobiliser. This step is intended as a time-saving mechanism so that transmission errors can be detected without completion of the whole authentication protocol.
5. The transponder recovers the nonce $X = R_{k_G}(Y)$ by applying the nonce-recovery algorithm to the challenge Y.
6. The immobiliser and the transponder both compute the AUT64 encryption of the nonce X, using the key K, and then optionally compress the result by adding the bottom and top halves of the ciphertext together modulo 2. The transponder sends the encrypted nonce to the immobiliser which then compares the response with the locally computed encryption. If the bit strings match then the authentication succeeds and the vehicle is mobilised.

The challenge algorithm C is based on a proprietary LFSR-based stream cipher that takes as input the compression function key part k_G. More precisely, the challenge cipher

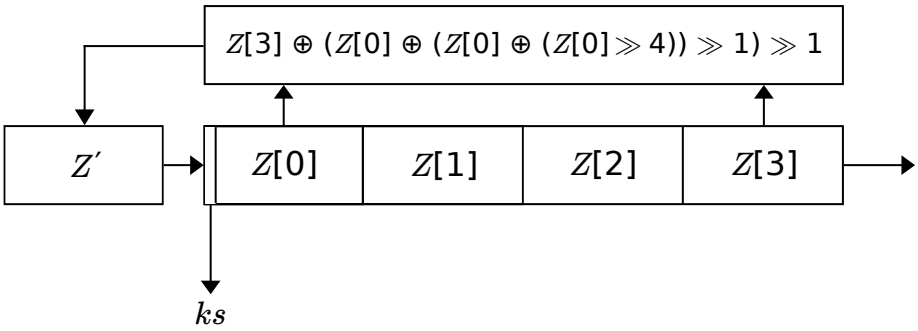


Figure 4.4: The proprietary TK5561 challenge LFSR cipher C.

state is a 32-bit register Z that is seeded with a value derived from k_G as follows:

$$Z = k_G[0] \oplus \texttt{0xD5} \parallel k_G[1] \oplus \texttt{0x89} \parallel k_G[2] \oplus \texttt{0x0C} \parallel k_G[3] \oplus \texttt{0x7B}$$

Once seeded, the challenge cipher is used to generate a 64-bit keystream that is used to compute the challange C by adding the keystream, bitwise and modulo 2, to the nonce X. Each keystream bit is computed as follows. First, the algorithm calculates the feedback-byte Z' from the register state Z such that:

$$Z' = Z[3] \oplus \Big(Z[0] \oplus \Big(Z[0] \oplus (Z[0] \gg 4) \Big) \gg 1 \Big) \gg 1$$

Next, the rightmost bit of the feedback byte Z' is shifted into the LFSR at position zero and is then output as the next key stream bit ks.

In the following sections we first identify a number of theoretical, cryptographic weaknesses in the design of AUT64 as a symmetric encryption primitive. Then, in Section 4.7, we focus on the concrete immobiliser system implementation and present two practical attacks on the use of AUT64 within a complete system.

4.5 Weak Keys

Ideally, for all AUT64 key pairs $K_0, K_1 \in \langle k_G, k_\sigma, k_\tau \rangle$, there should be no simple relation between the AUT64 encryption functions $\text{AUT64}_{K_0}(\cdot)$ and $\text{AUT64}_{K_1}(\cdot)$. In this section, we identify two classes of weak keys [19] for which the AUT64 encryption function is weaker than the others. In both cases, we develop a chosen-plaintext membership test that succeeds in identifying the class of weak keys and that consequently reduces the effective key space.

4.5.1 Permutation Key Part

In Section 4.4.2, we introduce the detailed Feistel network structure of the AUT64 block cipher and show that the permutation key part k_σ is used to permute the cipher state in each round. For each bit in the plaintext to be replaced with the encrypted output from the AUT64 Feistel function, the cipher must be run for at least 8 rounds and the byte permutation σ_{byte}, defined by k_σ, must not have any fixed points. If σ_{byte} has a cycle length of less than 8 then the AUT64 ciphertext will always contain at least one byte from the original plaintext, regardless of how many rounds of encryption are applied. A chosen-plaintext adversary can identify weak permutation keys by asking for a small number of encryptions and inspecting the resulting ciphertexts.

Since any cipher that leaves whole bytes of the encrypted plaintext unchanged in the ciphertext is trivially insecure, k_σ contains a class of weak keys that must be avoided. The byte permutation σ_{byte} defined by k_σ must be a cyclic permutation of length 8, with no fixed points. Placing these structural constrains on k_σ considerably reduces the effective key length from the 24 bits as stated in the patent [34]. To begin with, restricting k_σ to only bijective permutations reduces the key space from 2^{24} to just $8! \approx 2^{15.3}$. Introducing the additional limitation that only cyclic permutations are permissible, the remaining effective key space of k_σ is just $(8-1)! \approx 2^{12.3}$. In general, if k_σ is cyclic and n elements

of the permutation are known then the number of possible key values is $(7-n)!$ [24].

4.5.2 Compression Function Key Part

In accordance with Definition 4.4.5, the AUT64 compression function G takes as input the permuted cipher state $x'_0, \ldots, x'_7$, the key part k_G, the round number r and the three key-independent lookup tables T_{offset}, T_U and T_L. As noted in Section 4.4.2, whilst an ideal compression function would output a uniformly random 4-bit value, G exhibits non-uniform behaviour when either the compression function key part k_G or the input state $x'_0, \ldots, x'_7$ contains a nibble with the value zero.

Since any k_G nibble with the value zero causes T_{offset} to output zero, regardless of the corresponding input nibble being compressed, these keys are undesirable and cause the strength of the AUT64 encryption algorithm to be weakened. In the most extreme case, where all of the nibbles in k_G have the value zero, the entire AUT64 encryption algorithm is reduced to a constant value. Correspondingly, the compression function key parts k_G that contain one or more nibbles with the value zero constitute a class of weak keys. We note that for different applications of AUT64, this class of weak keys may be undesirable for a second reason. When the round key nibble selected from k_G has the value zero, it is no longer possible to use the output from T_{offset} to recover the corresponding input nibble, AUT64 is no longer bijective and decryption is not possible. Removing the weak keys in k_G that contain zero nibbles reduces the effective key length $|k_G|$ to $15^8 \approx 2^{31.3}$.

We now present a chosen-plaintext attack on 8 round AUT64 that identifies certain weak compression function keys k_G. In relation to the key-independent lookup tables T_U and T_L provided in Appendix A, the 3rd and 6th nibbles of k_G are of particular significance. In the first round of AUT64 encryption, $k_G[3]$ and $k_G[6]$, are used to encrypt the upper and lower nibble, respectively, of the last input byte x'_7. After 8 rounds of AUT64 encryption, the output of the first round is present in the ciphertext for analysis. In the following

chosen-plaintext attack, we exploit the structure of the compression function to determine whether $k_G[3] \stackrel{?}{=} 0$, $k_G[6] \stackrel{?}{=} 0$ and $k_G[3] \stackrel{?}{=} k_G[6]$. We formulate our attack as the following game played between an adversary $\mathcal{A}$ and the AUT64 encryption oracle $\mathcal{O}^{\text{AUT64}}$:

1. The adversary $\mathcal{A}$ submits the plaintext $(\texttt{0x00})^8$ to the oracle $\mathcal{O}^{\text{AUT64}}$ and learns the corresponding ciphertext $c_{\text{ref}} = \mathcal{O}^{\text{AUT64}}(\texttt{0x00})^8$.
2. $\mathcal{A}$ generates the following set of plaintexts that will be used to determine whether $k_G[3] \stackrel{?}{=} 0$. First, the adversary $\mathcal{A}$ chooses a fixed nibble value n. $\mathcal{A}$ then composes the set of 8 plaintexts $P_{k_G[3]}$ such that each plaintext has 7 bytes with the value `0x00` and one byte with the value `0x0n`. Each plaintext should have the target byte `0x0n` at a different position, e.g. $P_{k_G[3]} =$

   (`0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0n`)

   (`0x00,0x00,0x00,0x00,0x00,0x00,0x0n,0x00`)

   ...

   (`0x0n,0x00,0x00,0x00,0x00,0x00,0x0n,0x00`)

3. The adversary $\mathcal{A}$ submits the plaintexts in $P_{k_G[3]}$ to the AUT64 encryption oracle $\mathcal{O}^{\text{AUT64}}$ and learns the corresponding ciphertext set $C_{k_G[3]}$. If any of the ciphertexts in $C_{k_G[3]}$ match the reference ciphertext c_{ref} obtained in the first step of the game then $\mathcal{A}$ learns that $k_G[3] = 0$.
4. $\mathcal{A}$ repeats a similar process to determine whether $k_G[6] \stackrel{?}{=} 0$. Once again, $\mathcal{A}$ chooses a fixed nibble value n and then composes the set of 8 plaintexts $P_{k_G[3]}$ such that each plaintext has 7 bytes with the value `0x00` and one target byte. This time, the target byte `0xn0` has the fixed value n located in the upper nibble position.
5. As earlier, the adversary $\mathcal{A}$ submits the plaintexts in $P_{k_G[6]}$ to the AUT64 encryption oracle $\mathcal{O}^{\text{AUT64}}$ and learns the corresponding ciphertext set $C_{k_G[g]}$. If any of the

ciphertexts in $C_{k_G[6]}$ match the reference ciphertext c_{ref} obtained in the first step of the game then $\mathcal{A}$ learns that $k_G[6] = 0$.

6. Finally, to determine whether $k_G[3] \stackrel{?}{=} k_G[6]$, $\mathcal{A}$ can compare the ciphertext sets $C_{k_G[3]}$ and $C_{k_G[6]}$. If the sets contain pairs of ciphertexts, c_3 from $C_{k_G[3]}$ and c_6 from $C_{k_G[6]}$, such that for some nibble values $a, b, c \in \{0,1\}^4$, c_3 always contains a ciphertext byte $b_3 = \texttt{0xab}$ and c_6 always contains a ciphertext byte $b_6 = \texttt{0xca}$, and both b_3 and b_6 are at the same position in c_3 and c_6, respectively, then $k_G[3] = k_G[6]$.

The intuition behind this attack is that when the input nibble has the value `0x0` then the value in the XOR sum output by the AUT64 compression function G is unchanged. In this attack, 15/16 of the input nibbles have the value `0x0` and so the output of the compression function in the first round is entirely determined by one nibble from the compression function key part k_G. Specifically, from the round key lookup tables T_U and T_L provided in Appendix A, the output is determined by either the 3rd or 6th nibble in k_G. If either $k_G[3] = 0$ or $k_G[6] = 0$ then, because the first row in T_{offset} contains only zeroes, encrypting one of the plaintexts in $P_{k_G[3]}$ or $P_{k_G[6]}$, respectively, will be identical to encrypting the reference string $(\texttt{0x00})^8$.

If either $k_G[3] = 0$, $k_G[6] = 0$ or $k_G[3] = k_G[6]$, the entropy of the compression function key part is reduced by approximately 4 bits. In the worst case, $k_G[3] = 0$ and $k_G[6] = 0$ and the adversary learns 8 bits of k_G.

4.6 Cryptanalysis of AUT64

In this section, we present our cryptanalysis of the AUT64 block cipher, identify several weaknesses in the design and formulate a number of theoretical attacks. Generally, the security of a block cipher would be evaluated by measuring its resistance to both linear [171] and differential [17] cryptanalysis. In particular, the substitution and permutation operations are often scrutinised to determine the presence of any differential covariance

or linearity [125]. In this work, owing the fact that in AUT64 the primary substitution and permutation operations are key-dependent, we seek to develop alternative techniques that provide general attacks which are independent of the key. Our cryptanalysis is predominantly applicable to the 8 round AUT64 configuration, and treats AUT64 as an encryption oracle in the chosen-plaintext setting.

4.6.1 Permutation Weakness

In this section we identify four weaknesses in the design of AUT64 and develop a corresponding attack that enables an adversary to determine an element from the permutation key part k_σ. Specifically, we present a 16 chosen-plaintext attack that reduces the effective key space of k_σ to just $(7-1)! \approx 2^{9.5}$. Our attack exploits the following four weaknesses:

1. The compression function G does not behave like a uniformly random oracle. Carefully chosen inputs can be used to cause highly distinguishable outputs.
2. A chosen-plaintext adversary can tightly control the input to G in the first round of encryption.
3. The Substitution-Permutation Network (SPN) that takes the output of G and completes the Feistel function F has a cryptographic weakness.
4. After 8 rounds of AUT64, the output from the Feistel function F in the first round is a byte in the ciphertext.

We formulate our attack as the following game played between an adversary $\mathcal{A}$ and the AUT64 encryption oracle $\mathcal{O}^{\text{AUT64}}$.

1. The adversary $\mathcal{A}$ generates the set of 16 chosen plaintexts P such that for every nibble value $n \in \{0, \ldots, 15\}$, there is a corresponding plaintext $p_n \in P$ that comprises 8

identical and symmetric bytes $p_n = (n \parallel n)^8$. In other words $P =$

$$(\texttt{0x00},\texttt{0x00},\texttt{0x00},\texttt{0x00},\texttt{0x00},\texttt{0x00},\texttt{0x00},\texttt{0x00})$$

$$(\texttt{0x11},\texttt{0x11},\texttt{0x11},\texttt{0x11},\texttt{0x11},\texttt{0x11},\texttt{0x11},\texttt{0x11})$$

$$\ldots$$

$$(\texttt{0xFF},\texttt{0xFF},\texttt{0xFF},\texttt{0xFF},\texttt{0xFF},\texttt{0xFF},\texttt{0xFF},\texttt{0xFF})$$

2. $\mathcal{A}$ submits the plaintexts $p \in P$ to the AUT64 encryption oracle $\mathcal{O}^{\text{AUT64}}$ and learns the corresponding ciphertext set $C = \{\mathcal{O}^{\text{AUT64}}(p) : p \in P\}$.
3. The adversary A creates 8 lists $C'_0, \ldots, C'_7$. For each byte index b in each ciphertext $c \in C$, A adds the byte $c[b]$ to the list C'_b.
4. For each list of bytes $C'_0, \ldots, C'_7$, A counts the total number of symmetric bytes. The adversary $\mathcal{A}$ guesses that the permutation key part $k_\sigma[b]$, that corresponds to the first round of AUT64 encryption, has the value 7.

This attack works by exploiting, with chosen plaintexts, weaknesses in the AUT64 compression function G that cause the output to always be a symmetric byte. The 16 chosen plaintexts in this attack force G to output the full range of 16 symmetric bytes $\texttt{0x00}, \ldots, \texttt{0xFF}$. Outputs from G are then substituted nibble-wise by the substitution operation S and the symmetry remains. Only the bitwise permutation σ_{bit} changes the symmetry of the byte before it is once again substituted with the same symmetry-retaining operation S. In more detail, the attack works as follows.

Each chosen $p \in P$ is first input to the byte permutation operation σ_{byte}. Since each plaintext is composed from 8 identical bytes, σ_{byte} has no effect. Choosing plaintexts with this structure allows the adversary to tightly control the input to the compression function G in the first round.

Since the compression function G is provided with an input composed from 8 identical

bytes, the adversary can be certain that this will cause G to output a symmetric byte. This property holds because G outputs a nibble-wise XOR sum of values derived from each byte in the input and the compression function key part k_G. The upper and lower nibble in the output sum is derived from the upper and lower nibbles in the input bytes, respectively. Each value in the sum is chosen from the key-independent lookup table T_{offset} using a nibble chosen from k_G and a nibble from the input. As all of the input nibbles have the same value, and each of the key nibbles from k_G are used in both the upper and lower sum, the same terms are always chosen from T_{offset}. The order of the terms will differ in the upper and lower parts of the output sum, but since addition modulo 2 is commutative, the final value will always be the same.

Finally, the symmetric byte output from G is input to the SPN that completes the Feistel function F and produces the round output. The SPN is an entirely key-dependent construction that uses both the substitution key part k_τ and the permutation key part k_σ. The only operation in the SPN that does not preserve the symmetry of the input is the bitwise permutation σ_{bit}. On average σ_{bit} will break the symmetry of the input byte, however there are at least two inputs that will be unaffected, regardless of the AUT64 key. At the least, symmetry of the SPN input byte will be preserved when the input to σ_{bit} is `0x00` and when it is `0xFF`. No matter how these bits are permuted the resulting byte will always have the same value. This property means that, for a minimum of 2 of the 16 chosen plaintexts, the output from the first round of AUT64 encryption will always be a symmetric byte. In the next section, we explore the full distribution of attack outcomes under different key values and use the results to further reduce the security of AUT64.

An adversary who successfully identifies one element of the permutation key part k_σ using this attack reduces the permutation key space from $7! \approx 2^{12.3}$ for a key that is known to be cyclic, to $(7-1)! \approx 2^{9.5}$ for a key where one element of the permutation is known.

4.6.2 Compression Function Weakness

In this section, we continue to develop the previous attack by considering that the output from the first round of encryption can be used to gain an advantage in determining the compression function key part. We reduce the security of AUT64 to less than 76.9 bits and show that for a class of weak keys, the entropy is as little as 44.7 bits.

When the adversary in Section 4.6.1 asks for the encryption of the set of chosen plaintexts $P = \left\{(n \parallel n)^8 : n \in \{0, \ldots, 15\}\right\}$, the corresponding set of compression function output bytes in the first round belong to one of 16 equivalence classes. This can be understood in terms of the compression function lookup table T_{offset}, which has 16 rows and columns. Each chosen plaintext $p \in P$ fixes one column in the table, leaving just 16 possible key-dependent outputs. In this attack, each AUT64 compression function key part k_G belongs to an equivalence class that is determined by the XOR sum of its nibbles. In other words, $\forall k_G \in \mathbb{F}_2^{32}$, $\mathsf{class}(k_G) = \bigoplus_{i=0}^{7} k_G[i]$.

The compression function output bytes that correspond to each key class can be read from T_{offset} in Appendix A by using the class membership function $\mathsf{class}(k_G)$ to select a row from the table. Considering the set of bytes output by the compression function G during the first round of encryption of the set of plaintexts P, there are two possible distributions:

1. Typically, G will output a uniform distribution comprising the 16 different nibble values. This is the expected behaviour, it means that G is behaving injectively and that decryption is feasible. Only 15/16 compression function key parts have this property so the key space is reduced from 2^{32} to $(\frac{15}{16} \times 15^8) \approx 2^{31.2}$.
2. For every 1/16 compression function keys k_G that belongs to the class $\mathsf{class}(k_G) = 0$, G does not behave injectively its output is reduced to a constant value. Correspondingly, this distribution can be used to reduce the effective key space of k_G to 2^{28}. If we consider only non-weak k_G as defined in Section 4.5.2, the key space is further

reduced to $(\frac{1}{16} \times 15^8) \approx 2^{27.3}$.

The compression function output distribution is preserved by the SPN that completes the AUT64 Feistel function F. Correspondingly, when considering the set of bytes C'_i output in the first round when encrypting the set of plaintexts P, either a uniform or a degenerate distribution will be formed. The distribution formed in C'_i can be used to determine some bits from the substitution key part k_τ.

On average, C'_i will have a uniform distribution and will also feature at least two symmetric bytes. The precise number of symmetric bytes is determined by the bitwise permutation σ_{bit}. We analysed the complete permutation key space to develop the following statistical distinguisher based on the number of symmetric bytes in the set C'_i. 46%, 42%, 11% and 1% of k_σ key values[1] result in exactly 2, 4, 8 and 16 symmetric bytes in C'_i, respectively.

The average case, that there are exactly two symmetric bytes in C'_i, is the second best outcome for the adversary. The adversary can be certain that these two bytes reveal the key-dependent substitutions of `0x00` and `0xFF`, as these values always remain constant despite permutation. When there are two symmetric bytes in C'_i, the security of the substitution key part k_τ is reduced from $16! \approx 44.3$ bits to just $2 \times 14! \approx 37.3$ bits. In addition, the security of the permutation key part k_σ and the compression function key part k_G is reduced to approximately $0.46 \times (7-1)! \approx 8.4$ bits and 31.2 bits, respectively. In total, the security of AUT64 is reduced to no more than $2^{76.9}$ possible key values.

The best case for an adversary is that encrypting the set of chosen plaintexts P produces 16 symmetric bytes in C'_i. This result reduces the entropy of k_σ to $48 \approx 5.6$ bits, of k_τ to $15 \times 16 \approx 7.9$ bits and of k_G to ≈ 31.2 bits. For this class of permutation keys, the security of 8 round AUT64 encryption is reduced to less than $2^{44.7}$ key values.

We note that sometimes a ciphertext byte index other than the one corresponding to

[1]Specifically, there are 2304, 2112, 576 and 48 cyclic k_σ values that produce 2, 4, 8 and 16 symmetric bytes in C'_i respectively.

the first round will also feature a number of symmetric bytes. In this case it may not be possible for the adversary to determine which set of ciphertext bytes C'_i corresponds to the output from the first round. To overcome this situation we developed a second technique. The adversary simply divides each candidate byte list $\mathbb{C}'_i$ into an upper and lower nibble set. The first round can always be uniquely distinguished by equality of the upper and lower nibble sets.

4.6.3 Compression Function Divide-and-Conquer

In this section we identify a second weakness in the AUT64 compression function G and develop a corresponding attack that reduces the security of AUT64 to no more than 57.7 encryptions. The way in which G makes nibble-wise use of the compression function key part k_G can be exploited to cause outputs that are dependent on just a single nibble in k_G. In each round, the key-independent lookup tables T_U and T_L are used to determine the permutation of k_G that is used to encrypt each upper and lower input nibble, respectively. If all of the input nibbles except one have the value zero, then the output of G in the first round will always be dependent on a single nibble in k_G. We exploit this behaviour with the following chosen-plaintext attack which we formulate as a game played between an adversary $\mathcal{A}$ and the AUT64 encryption oracle $\mathcal{O}^{\text{AUT64}}$:

1. The adversary A runs the 16 chosen-plaintext attack developed in Section 4.6.1 to determine the ciphertext byte position b_0 that corresponds to the output from the first round of encryption.
2. Next, A creates the set of 16 chosen plaintexts P such that each plaintext $p \in P$ comprises a unique nibble value $n \in \{0, \ldots, 15\}$, and 15 nibbles with the value zero. In each plaintext, n must be placed at the byte position b_0 that will be permuted by the byte permutation σ_{byte} into the 8th byte in the first round. For example where

$b_0 = 3$, $P =$

(0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00)

(0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00)

...

(0x00,0x00,0x00,0x0F,0x00,0x00,0x00,0x00)

3. The adversary A submits the set of chosen plaintexts P to the AUT64 encryption oracle $\mathcal{O}^{\text{AUT64}}$ and learns the corresponding ciphertext set $C = \{\mathcal{O}^{\text{AUT64}}(p) : p \in P\}$. A creates the set C'_{r_o} that comprises the ciphertext byte at position r_0 for every ciphertext $c \in C$. i.e. $C'_{r_o} = \{c[r_0] : c \in C\}$.
4. A now exhaustively searches the remaining key space that comprises the target compression-function key nibble $k_G[3]$, the permutation key part k_σ and the substitution key part k_τ. For each candidate key value $K \in \langle k_G[3], k_\sigma, k_\tau \rangle$, the adversary A computes the set of ciphertexts $C_{\text{ref}} = \{\text{Enc}_{\text{AUT64}}(K, p) : p \in P\}$ and the corresponding the set of bytes from the first round $C'_{\text{ref}} = \{c_{\text{ref}}[r_0] : c_{\text{ref}} \in C_{\text{ref}}\}$. The correct key has been found once $C'_{\text{ref}} = C'_{r_o}$.
5. The adversary A efficiently determines the remaining unknown key nibbles in k_G by supplying additional sets of chosen plaintext to the oracle $\mathcal{O}^{\text{AUT64}}$ that target different key nibbles. The simplest example is that by moving the nibble value n from the lower to the upper 4-bits at position r_0 in each ciphertext, the 6th key nibble $k_G[6]$ can be determined. Since the other key parts have already been determined, each unknown key nibble will only require 15 chosen plaintexts and up to 15 offline encryptions.

The attack presented above requires 137 chosen plaintexts and reduces the security of AUT64 to $(16! \times 6! \times 15) + (15 \times 7) \approx 2^{57.6}$ encryptions. The attack implementation

can be optimised significantly by observing that the output of the compression function G in the first round is only determined by a single row in T_{offset}. Indeed, G can be entirely reduced to T_{offset} and values from the table used directly to test candidate permutation and substitution key values. This is considerably less computation than running the full AUT64 encryption algorithm and $2^{57.6}$ operations is well within the reach of specialised setups for exhaustive key search [118].

4.6.4 Integral Cryptanalysis

In this section we generalise the permutation weakness attack from Section 4.6.1 and develop an integral cryptanalysis technique that enables an adversary to determine several permutation elements from the key part k_σ. Specifically, we present a 272 chosen-plaintext attack that reduces the cyclic permutation key space from 12.3 bits to just 6.9 bits.

Integral cryptanalysis [151] is a sibling of differential cryptanalysis that considers, rather than the propagation of differences between pairs of values, the propagation of sums of many values. Where differential cryptanalysis considers the XOR difference of plaintext-ciphertext pairs, integral cryptanalysis is concerned with sets of plaintexts, the corresponding sets of ciphertexts, and the XOR sums over the entire sets. Integral cryptanalysis is particularly effective against block ciphers that are composed from only bijective components.

The 8 round AUT64 implementation is vulnerable to the following chosen-plaintext integral cryptanalysis attack which we formulate as a game played between an adversary $\mathcal{A}$ and the AUT64 encryption oracle $\mathcal{O}^{\text{AUT64}}$. In this attack, $\mathcal{A}$ learns two elements from the permutation key part k_σ:

1. The adversary $\mathcal{A}$ runs the permutation weakness attack from Section 4.6.1. This attack requires 16 chosen plaintexts, negligible computation and determines the byte position r_0 that corresponds to the output from the first round of encryption.

2. $\mathcal{A}$ generates the set of 256 chosen plaintexts P such that each plaintext $p \in P$ has 7 bytes with the value `0x00` and one byte with a unique 8-bit value $n \in \{\texttt{0x00}, \ldots, \texttt{0xFF}\}$. In each plaintext p, n is placed at the position r_0 so that in the first round, the permuted plaintexts input to the compression function are of the form $(\texttt{0x00})^7 \parallel n$. For example if $r_0 = 3$, then $P =$

   (0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00)

   (0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00)

   ...

   (0x00,0x00,0x00,0xFF,0x00,0x00,0x00,0x00)

3. The adversary A submits the set of chosen plaintexts P to the AUT64 encryption oracle $\mathcal{O}^{\text{AUT64}}$ and learns the corresponding ciphertext set $C = \{\mathcal{O}^{\text{AUT64}}(p) : p \in P\}$. A creates the ciphertext byte sets $C'_0, \ldots, C'_7$ such that each set C'_i comprises the ith ciphertext byte $c[i]$ from every ciphertext $c \in C$.
4. To complete the attack, A computes the XOR sum of each ciphertext byte set C'_i such that $\forall i \in \{0, \ldots, 7\}, \forall c_j \in C'_i : \mathsf{sum}(C'_i) = \bigoplus_{j=0}^{255} c'_j$. The adversary will find that 2/8 of these sums will have the value `0x00` and that this also corresponds to the XOR sum of the plaintext set, $\mathsf{sum}(P)$. One of the sums with the value `0x00` will correspond to the set of ciphertext bytes output in the first round C'_{r_0}, the other will correspond to the output in the second round position r_1. The adversary A uses knowledge of the round positions r_0 and r_1 to reduce the entropy of the permutation key part k_σ from 12.3 bits to just 6.9 bits.

The intuition for why this attack works is to consider that because AUT64 is bijective, each round must also be bijective. This necessitates that each unique plaintext $p \in P$ is encrypted into a unique intermediate state in the first round, otherwise there would be

colliding outputs and it would not be possible to determine a unique decryption for each ciphertext. The set of chosen plaintexts P we describe in this attack forces the output from the first two rounds to always comprise the set of all possible byte values. For the nth plaintext $p_n \in P$, the attack works as follows:

1. The byte permutation σ_{byte} applied to p_n outputs a permuted state of the form $(\texttt{0x00})^7 \parallel n$.
2. The Feistel function F is applied to the permuted state and the resulting round output is $(\texttt{0x00})^7 \parallel F\Big((\texttt{0x00})^7 \parallel n\Big)$. Since every n is a unique byte value and to preserve bijectivity, every encrypted byte $F\Big((\texttt{0x00})^7 \parallel n\Big)$ must also be unique. The XOR sum of 256 unique byte values is always `0x00` and this allows the adversary to distinguish the output of the first round in this attack.
3. The second round of encryption begins and σ_{byte} is applied to the output from the previous round. The output of the Feistel function in the first round $F\Big((\texttt{0x00})^7 \parallel n\Big)$ is permuted to the byte position that is determined by the permutation key part k_σ.
4. The Feistel function F is applied again to a state that comprises 7 bytes with the value `0x00` and one byte with the value $F\Big((\texttt{0x00})^7 \parallel n\Big)$. Since every state $F\Big((\texttt{0x00})^7 \parallel n\Big)$ is unique and to preserve bijectivity, the resulting encrypted byte in the second round must also have a unique value. The zero-value XOR sum of the 256 unique encrypted bytes allows us to distinguish the output of the second round in this attack.
5. The third round of encryption begins and σ_{byte} is applied to the output from the previous round. This time, there are two non-zero values in the state that correspond to the outputs from the first and second round of encryption. This time, the output of the Feistel function F when applied to the permuted state does not have to be unique as long as the overall resulting 64-bit state is. The XOR sum of the ciphertext bytes corresponding to the third round are therefore unlikely to have the value `0x00`.

Encryption continues in this way for another 5 rounds.

This attack requires a total of 272 chosen plaintexts, and reduces the entropy of the permutation key part to just 5 unknown elements. Applying the calculation from Section 4.5.1, the resulting effective permutation key space is just $(7-2)! \approx 2^{6.9}$ values.

4.6.5 Extended Integral Cryptanalysis

Finally, in this section we fully generalise the integral cryptanalysis attack and determine its limitations. We present a chosen-plaintext attack that determines the third and fourth-round permutation elements and that correspondingly reduces the security of the permutation key part to just 6 possible values.

The intuition for this attack is that permutation elements from the AUT64 key can be iteratively identified by adaptively formulating chosen-plaintext attacks that cause a zero-sum ciphertext byte set to propagate into the target round. In Section 4.6.4, a chosen-plaintext set P is developed that propagates an XOR sum of `0x00` into the first and second round ciphertext byte sets. To cause propagation of the distinguishing XOR sum into the third round, it is necessary to cause an intermediate state in the second round in which the two non-zero bytes have the same value. Under this condition, the preservation of bijectivity will force the output byte set of the third round to be a uniform byte distribution with an XOR sum value of `0x00`.

Firstly, we extend the attack in Section 4.6.4 to propagate the distinguishing zero-value XOR sum into the third round ciphertext byte set C'_{r_2}. The chosen plaintexts that are required for this attack are composed from 6 bytes with the value `0x00` and two byte counters $n, m \in \{0, \ldots, 255\}$. The byte counters n and m should be placed at the byte positions corresponding to rounds one r_0 and two r_1, respectively. There are 2^{16} such

plaintexts, for example if $r_0 = 3$ and $r_1 = 5$ then $P =$

(0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00)

(0x00,0x00,0x00,0x00,0x00,0x01,0x00,0x00)

...

(0x00,0x00,0x00,0x00,0x00,0xFF,0x00,0x00)

(0x00,0x00,0x00,0x01,0x00,0x01,0x00,0x00)

(0x00,0x00,0x00,0x01,0x00,0x02,0x00,0x00)

...

(0x00,0x00,0x00,0xFF,0x00,0xFF,0x00,0x00)

From the total set of 2^{16} chosen plaintexts in P, only a small subset of size 256 will cause the necessary intermediate states at the output of the second round. A general method for identifying the set of plaintexts P' that propagate the ciphertext set byte sum 0x00 into the third round is to exhaustively encrypt the entire set of plaintexts P. P' is the subset of P that encrypt to the ciphertexts $c \in C$ such that $c[r_0] = c[r_1]$. The ciphertexts corresponding to the encryptions of $p \in P$ should be added to the set C' that will be used to identify the third round permutation element r_2.

To identify position r_2, we create the ciphertext byte sets $C'_0, \ldots, C'_7$ such that each set C'_i comprises the ith ciphertext byte $c[i]$ from every ciphertext $c \in C'$. Computing the XOR sum of each byte set $\mathsf{sum}(C'_i)$ will reveal 3/8 sets with the sum 0x00. Eliminating the known indices of the first and second rounds will reveal the byte set C'_{r_2} that corresponds to the output of the third round of encryption. Knowledge of r_0, r_1 and r_2 reduces the entropy of the permutation key part to just 4 elements and leaves an effective permutation key space of just $(7-3)! = 24$ possible values.

This attack can be extended to the fourth round as follows. As before, the objective is

to cause a zero-sum ciphertext byte set to propagate into the target round. To accomplish this, the desired output state in the third round has 5 bytes with the value `0x00` and 3 encrypted bytes that have the same value. Bijectivity necessitates that this intermediate state will cause, for each unique plaintext, a unique ciphertext byte to be output in the fourth round. The exhaustive plaintext set for this attack is generated by placing three distinct byte counters at the positions r_0, r_1 and r_2. The resulting set of 2^{24} plaintexts P are encrypted with AUT64 and then, as before, the subset P' that cause the fourth round to be distinguishable is created by identifying the corresponding ciphertext set $C' = \{c[r_0] = c[r_1] = cr_2 : \forall c \in C\}$. Once again, the ciphertext byte sets $C'_0, \ldots, C'_7$ are composed from the ith ciphertext byte $c[i]$ from every ciphertext $c \in C'$ and the XOR sum of each byte set is calculated. Eliminating the sums that correspond to the known byte indices of the first, second and third rounds will reveal the byte set C'_{r_3} that corresponds to the output of the fourth round of encryption.

Determining the first four elements from the permutation key part k_σ requires a total of $272 + 2^{16} + 2^{24} \approx 2^{24}$ chosen-plaintext encryptions and reduces the security of k_σ to just 6 possible values. The limitation of this technique is that the number of chosen-plaintext encryptions grows exponentially in the number of permutation elements that are revealed. In practice, it is best to use the 272 chosen-plaintext attack in Section 4.6.4 to reduce the entropy of k_σ to 120 possible key values and then to continue the search offline.

4.6.6 Beyond 8 Rounds

The cryptographic weaknesses presented in this section are all dependent on the adversary being able to cause the Feistel function to have a distinguishable output in the target round. Beyond 8 rounds of encryption, AUT64 begins to overwrite the ciphertext bytes that were output in earlier rounds. In particular, the output of the 9th round overwrites the output from the 1st round, the 10th round output overwrites the 2nd round, and so

on. In general, the attacks in this section can distinguish $16 - N$ permutation elements from N round AUT64.

For example, to attack the 5th round of the 12 round AUT64 implementation that was found in the VW Group RKE system evaluated by Garcia et al. [105], it is necessary to determine exhaustively the byte counter positions that correspond to the first 4 rounds of encryption. The attack requires $\binom{8}{4} \times 2^{32} \approx 2^{38.1}$ chosen plaintexts and reduces the cyclic permutation key space from 12.3 bits to $4! \times 2! \approx 3.6$ bits.

4.7 Attacking the AUT64 Implementation

In this section, we present two practical attacks on the use of AUT64 within a real immobiliser system. First we evaluate the key derivation implementation, identify a significant weakness and present a corresponding attack that breaks 24 round AUT64. Second, we outline a very fast attack on 8 round AUT64 when k_G is known.

4.7.1 Key Derivation Weaknesses

Based on the Mazda immobiliser system that we reverse engineered, we found that the immobiliser unit derives the 32 bit compression function key part k_G from the transponder ID code that is transmitted each time the vehicle key is energised by the immobiliser antenna. This functionality is not indicated in the AUT64 documentation [34] and represents a significant vulnerability. The key part k_G is derived as follows:

Definition 4.7.1 (Compression Function Key Derivation) Let ID_0, ID_1, ID_2, ID_3 denote the 4 bytes that comprise the transponder ID code and let $u = (ID_0 \wedge \texttt{0xE}) \ll 1$, then where T_D is the key-derivation lookup table in Appendix A, each byte in k_G is derived

as follows:

$$k_{G_0} = ID_0 \oplus T_D[u] \oplus k_{G3}$$
$$k_{G_1} = ID_1 \oplus T_D[1+u] \oplus k_{G3}$$
$$k_{G_2} = ID_2 \oplus T_D[2+u] \oplus k_{G3}$$
$$k_{G_3} = ID_3 \oplus T_D[3+u]$$

The derivation of k_G from the transponder ID code and the lookup table T_D is an entirely key-independent operation. This means that the adversary can trivially compute 32 bits of the AUT64 key k_G with just a single communication with the transponder. Not only does this reduce the security of AUT64 by 32 bits, but as we present in the following sections, it also weakens the security of the remaining key parts.

4.7.2 24 Rounds

The Mazda immobiliser system that we evaluate in this work uses AUT64 in the 24 round configuration. To recover the permutation and substitution key parts, k_σ and k_τ respectively, we first recover the compression function key part k_G by energising the transponder which then sends its ID code. Assuming that all 4×4 bijective S-Boxes and all cyclic permutations are candidates, the remaining AUT64 key space is $16! \times 7! \approx 2^{56.5}$ and within the scope of dedicated exhaustive search devices [118]. However, the AUT64 key management scheme specified in the patent [34] makes it clear that the permutation key space is reduced to just 16 keys per vehicle manufacturer. This means that by reading k_σ from two different immobiliser units and identifying the constant part, the AUT64 key space is no more than $16! \times 16 \approx 2^{48.3}$. Once the permutation key part is known for a particular manufacturer's vehicle family, the key space is only $16! \approx 2^{44.3}$ and may be found using exhaustive search in minutes using parallel GPU-based Amazon EC2 instances

[14].

4.7.3 8 Rounds

In this final section, we give an example of exploiting the cryptographic weaknesses in Section 4.6 in a practical setting. We assume the use of an 8 round AUT64 implementation, that the compression function key part k_G can be derived as in Section 4.7.1 and that we obtain all 8 ciphertext bytes. Note that in the TK5561 authentication protocol, we receive the 32-bit XOR sum of the upper and lower ciphertext halves, while the patent specifies this reduction as optional. We leave the extension of the following attack to the case when only the XOR is available for future work. Under the above assumptions, we develop an attack that typically breaks 8 round AUT64 within milliseconds using a standard laptop. We formulate our attack as the following game played between an adversary $\mathcal{A}$ and the AUT64 encryption oracle $\mathcal{O}^{\text{AUT64}}$:

1. The adversary $\mathcal{A}$ uses the 256 chosen-plaintext integral cryptanalytic attack in Section 4.6.4 with the following set of chosen plaintexts

$$P_0 = \left\{ \left((\texttt{0x00})^7 \parallel n \right) : n \in \{0, \ldots, 255\} \right\}$$

   Placing the byte counter n at position 7 in each plaintext guarantees that only the first round permutation element r_0 is identified. $\mathcal{A}$ submits the plaintexts $p \in P_0$ to the AUT64 encryption oracle $\mathcal{O}^{\text{AUT64}}$ and receives the corresponding set of ciphertexts C_0. $\mathcal{A}$ creates the ciphertext byte sets $C'_{0,0}, \ldots, C'_{0,7}$ such that each set $C'_{0,i}$ comprises the ith ciphertext byte $c[i]$ from every ciphertext $c \in C_0$. $\mathcal{A}$ identifies the ciphertext byte set C'_{0,r_0} and the permutation element r_0 that correspond to the first round of encryption such that $\mathsf{sum}(C'_{0,r_0}) = 0$.
2. The adversary $\mathcal{A}$ uses the byte set C'_{0,r_0} to build a model of the unknown SPN that completes the AUT64 Feistel function F. The model is a 256 element lookup table

that has the same functionality as the key-defined SPN from Section 4.4.2. $\mathcal{A}$ is able to model the SPN because the input can be computed offline using the chosen plaintext input set P_0, the key part k_G and the compression function G. The model outputs are the corresponding ciphertext bytes in C'_{0,r_0}.

3. The following step is an optimisation that may be omitted depending on the ease of retrieving chosen plaintexts. The adversary $\mathcal{A}$ composes a second set of chosen plaintexts P_1 such that each plaintext comprises 7 bytes with the value zero and one byte with a unique byte counter value n. This is similar to the plaintext set used in the compression function divide-and-conquer attack in Section 4.6.3.

$$\mathbb{P}_1 = \left\{ (64 - (8 \times r_0) \ll n) : n \in \{0, \ldots, 255\} \right\}$$

   $\mathcal{A}$ submits the plaintexts $p \in P_1$ to the AUT64 encryption oracle $\mathcal{O}^{\text{AUT64}}$ and receives the corresponding set of ciphertexts C_1. As-per Step 1, $\mathcal{A}$ creates the ciphertext byte sets $C'_{1,0}, \ldots, C'_{1,7}$ and then identifies the sets of bytes C'_{1,r_0} and C'_{1,r_1} that correspond to the output of the first and second rounds of encryption, respectively.

4. The adversary $\mathcal{A}$ adaptively determines the permutation key part k_σ by using the SPN model to calculate the expected round outputs for each unknown permutation element. For example, to determine the permutation element corresponding to the third round we propose the following technique. $\mathcal{A}$ uses the ciphertext byte sets C'_{1,r_0} and C'_{1,r_1} from Step 3 to create the set of all possible intermediate cipher states X'_1 that could be input to the third round. $\mathcal{A}$ has to consider no more than $\binom{7}{2} = 21$ possible byte permutations. The adversary $\mathcal{A}$ computes the compression function for each possible intermediate state $x \in X'_1$ and then applies the SPN model from Step 2. The correct byte permutation will produce the set of output bytes that match the ciphertext byte set $C'_{1_{r_2}}$ corresponding to the output of round three.

This technique can be extended to determine additional permutation elements by computing the possible permutations of the relevant ciphertext byte sets. Once the first four permutation elements have been found using this attack, only 6 keys remain and it is more efficient to continue using exhaustive search.

5. Finally, the adversary $\mathcal{A}$ uses the following method based on the weaknesses identified in Section 4.6.2 and knowledge of the permutation key part k_σ, to very efficiently determine the remaining substitution key part k_τ. The compression function weakness from Section 4.6.2 determines that the set of ciphertext bytes $C'_{1_{r_0}}$ from Step 3. will always contain two or more symmetric bytes. $\mathcal{A}$ can use the symmetric bytes in $C'_{1_{r_0}}$, the model of the SPN from Step 2. and the partial knowledge of k_σ to vastly reduce the number of candidate substitution key values.

The worst case for an adversary running the attack above is that the remaining substitution key space is $2 \times 14! \approx 2^{37.3}$. On average, 10 or more of the substitution elements from k_τ will be confirmed and the remaining key space will have fewer than 150 candidates. Experimentally, we confirmed that the majority of AUT64 keys are broken within milliseconds on a standard Intel i7 laptop.

4.8 Chapter Summary

In this chapter, we present a complete analysis of the AUT64 automotive block cipher and its associated immobiliser protocol. We identify a number of cryptographic weaknesses in AUT64 and develop several attacks on both the 8 and 24 round implementations. Despite AUT64 having a 120 bit secret key length, we show that in certain implementations 8 round AUT64 can be broken within milliseconds using a standard laptop, with a worst-case complexity of $2^{37.3}$ encryptions. In the Mazda immobiliser system that we evaluate, the security of 24 round AUT64 is no more than 48.3 bits and can be exhaustively searched. We show that part of the secret key is actually derived from the transponder ID code and can be efficiently determined from only a single interaction with the transponder.

This chapter shows that it is imperative that the automotive industry discontinues the use of proprietary cryptographic implementations and that the vehicles of tomorrow are developed using standardised algorithms and peer-reviewed protocols. In addition to the immediate automotive impact, this work also provides a more general contribution to the literature on the cryptanalysis of generalised Feistel ciphers with key-dependent permutations and S-Boxes.

Chapter 5

Issue First Activate Later Certificates for V2X

In this chapter we present IFAL, a provably secure and privacy conscious scheme for V2X communication. Issue First Activate Later (IFAL) is a practical and secure improvement to the leading European standard for ITS, and one that also merits over the leading U.S. candidate. IFAL introduces a novel certificate issuance mechanism that both avoids the need for certificate revocation and that also improves privacy for vehicles with limited and intermittent connectivity. We present a new asymmetric key diversification technique that allows for certificate pre-issuance without the need for revocation. Pseudonym certificates are activated using small activation codes that enable vehicles to sign broadcast messages only during the activation period. We evaluate the security and privacy of IFAL in a formal game-based setting and show that our scheme is provably secure and privacy conscious. IFAL is efficient, standards-compliant and a good candidate for mainstream deployment.

5.1 Motivation

In V2X communication, road users are functionally required to broadcast unencrypted safety messages that precisely specify their current location, speed and heading. The precision and linkability of broadcast messages provides vehicles and infrastructure with a richer and more contemporary picture of the environment, enabling the creation of systems that facilitate safer and more efficient roads. At the same time it is critically important that drivers are protected from the type of long-term tracking that threatens to uniquely identify individual habits. However, road users must still be accountable for the messages they broadcast and it must be possible to revoke the credentials of misbehaving vehicles. These conflicting security and privacy requirements make the development of V2X schemes particularly challenging and has motivated considerable research activity.

For maximum impact on road safety, V2X should be universally deployed [138]. Correspondingly there are a number of international standardisation efforts. In particular there are two leading security architecture proposals for V2X. There is a European ETSI standard [85] and a U.S. DOT approach based on the Secure Credential Management System (SCMS) [29]. Both schemes are based on ECDSA digital signatures and a hierarchical PKI that manages certificates and provides trust between different road users. In addition, the IEEE Wireless Access in Vehicular Environments (WAVE) suite of standards provides the 802.11p physical layer specification and V2X broadcast message packet structure that unites the two ITS standards. In Europe, the adoption of these standards is strongly encouraged by Directive 2010/40/EU of the European Parliament [72] which mandates interoperable communication between connected vehicles.

In both the ETSI and U.S. DOT standards, security and privacy is managed using pseudonym certificates and a PKI that comprises a number of authorities. In particular, both schemes include at least one authority that issues long-term enrolment certificates and

also one or more authorisation authorities that issue short-term pseudonym certificates. The separation of enrolment and authorisation is a privacy mechanism that prevents any single authority from linking long-term vehicle behaviour to a specific vehicle owner. To protect vehicle owners from long-term tracking, vehicles use short-lived pseudonym certificates in combination with a pseudonym change strategy that ensures each journey comprises several different identities. Whilst pseudonym change alone is not enough to ensure unlinkability [259], privacy does correlate positively with the frequency of pseudonym change and it is desirable to provide vehicles with a large number of different identities [189].

To enable the high level of trust that is required for directing vehicle behaviour using V2X broadcast messages, it is critical that the credentials of vehicles which broadcast misleading information can be revoked. With around 300 million vehicles in Europe alone [3] and potentially thousands of pseudonym certificates per vehicle, revocation for vehicular networks is a substantial challenge. The traditional Certificate Revocation List (CRL) approach is not well suited to either the scale, mobility or latency requirements of V2X and this has motivated the development of a number of alternative techniques [213]. The two leading standards for ITS propose different revocation mechanisms. The ETSI approach is passive, vehicles are issued short-lived pseudonym certificates that are simply allowed to expire without replenishment when a vehicle is revoked. The U.S. DOT technique is based on a CRL, but the pseudonym certificates incorporate a linkage value that enables all of the certificates issued to a vehicle to be revoked with one CRL entry.

Neither of the revocation techniques in the standards are ideal. Short-lived certificates, as implemented in the ETSI architecture, are problematic for vehicles with limited or intermittent connectivity. When bandwidth is constrained, it is either necessary to issue certificates with relatively long validity periods or it necessary to pre-issue a large number of short-lived certificates. The first of these approaches reduces driver privacy because

the vehicle will be linkable over a long period of time and the second approach reduces trust in the messages sent by the vehicle as it cannot be revoked until the certificates have expired. The U.S. DOT approach is troublesome because vehicles are required to have a very high verification throughput [214] and must check the CRL before a message can be verified. IFAL is motivated by need to improve the standards with respect to revocation, in particular we seek to address the trade-off between privacy, trust and bandwidth introduced by the ETSI standard.

5.2 Contributions

IFAL is an improvement to the leading European security architecture candidate for ITS [85]. In IFAL vehicles are pre-issued with a lifetime supply of pseudonym certificates that are only usable upon receiving small, time-delayed activation codes. At any given point in time IFAL provides each vehicle with a unique and non-repeating pseudonym and, in contrast to the ETSI standard, provides location privacy that is not dependent on bandwidth or connectivity. In comparison to the U.S. DOT approach [29], IFAL offers improved and optimal Sybil attack resistance and avoids the need to reuse pseudonyms across multiple journeys. The contributions in this chapter are threefold

1. We present our new IFAL V2X scheme. IFAL is fully compliant with the latest ETSI security architecture standard [86] and includes several new features such as the ability to pre-issue pseudonym certificates that are only usable upon receiving small activation codes. IFAL has improved support for vehicles with limited connectivity, without reducing their privacy and furthermore avoids the need for certificate revocation which does not scale well and introduces significant message verification delays.
2. We provide the first formalisation of the security and privacy requirements set out in the ETSI ITS standard, in a provable security setting, and then show that IFAL

is a secure and privacy conscious V2X scheme.

3. We introduce a new asymmetric key-diversification technique with time-delayed activation that may have applications beyond V2X.

The remainder of this chapter is structured as follows. To begin, we briefly summarise our notation and then we review the standard requirements for V2X from Section 2.5.4, providing a more specific formulation. In Section 5.5 we introduce our system and attacker model and then, in Section 5.7, we present the full design and specification of our IFAL scheme. We present our formalisation of the ETSI V2X security and privacy requirements in Section 5.9 and our proof that IFAL is secure and privacy conscious in Section 5.9. Finally, we evaluate IFAL with respect to the standard requirements and present the results of our reference implementation in Section 5.10.

5.3 Notation

In this chapter we make use of the following notation conventions. We let $\texttt{enc}(k, m)$ and $\texttt{ENC}(P, m)$ denote the symmetric and public-key encryption of the arbitrary message m under the symmetric key k and public key P, respectively. We also let $\texttt{hash}(m)$ denote a secure hash function such as SHA-256 [193] applied to the message m. Where $\mathcal{K}$ is some keyspace we write $k \xleftarrow{\$} \mathcal{K}$ to denote choosing the symmetric key k uniformly at random from $\mathcal{K}$. To express the number of bits n in the bitstring k, we write $|k| = n$.

Concerning elliptic curve operations we let q be a prime or an order of 2, n be a prime and greater than 2^{160}, and then we let C denote an elliptic curve over the field $\mathbb{F}_q$. We use G to denote the point on the curve that generates a cyclic subgroup of order n under addition, such that the discrete logarithm problem in the subgroup spanned by G is hard. To distinguish between group and scalar multiplication we use '$\times$' and '$\cdot$' respectively.

In the formal security setting we use the term t to refer to some infeasible computational duration and the term ε to mean some negligible quantity such that t/ε is greater than

the running time of any feasible attacker. We use the term efficient to mean solvable using a probabilistic polynomial-time Turing machine with an error probability of less than 1/2.

Finally, we use the term "canonical identity" to refer to the unique vehicle registration information that would typically be managed by a national vehicle registration agency.

5.4 Requirements

In Section 2.5.4 we introduced the standard security, privacy and trust requirements for V2X that were identified by the European PRESERVE project [214] and which are the basis for both the latest ETSI and the U.S. DOT security architecture standards for ITS. In this section, we reformulate the same standard requirements in a more precise and symbolic setting. We let σ_i denote a digital signature with respect to the pseudonym public key P_i and the corresponding pseudonym certificate ρ_i. In other words, the arbitrary signed message tuple (m, σ_i) is valid with respect to the pseudonym public-key P_i and is authorised by the pseudonym certificate ρ_i. In this chapter, we define the standard security and privacy requirements for V2X as follows

Authenticity and Authorisation The main security requirement for a V2X scheme is that there is a mechanism for determining the authenticity and integrity of broadcast messages. The recipient of the signed message tuple (m, σ_i) and the corresponding pseudonym certificate ρ_i must be able to determine that the message m originates from the road user that the certificate ρ_i was issued to.

Pseudonymity Pseudonym certificates allow vehicles to broadcast signed messages without revealing their canonical identity. A signed message tuple (m, σ_i) and the corresponding pseudonym certificate ρ_i must not reveal the canonical vehicle identity.

Unlinkability The main mechanism for privacy in V2X is the use of multiple pseudonym

certificates such that every vehicle journey is divided into multiple identities. Vehicles should be able to repeatedly use the cooperative awareness service without an observer being able link the usage to a single source. Given the pair of pseudonym certificates ρ_i, ρ_j, an adversary should have no advantage in guessing whether the certificates originate from a single vehicle or not.

Accountability Vehicles must remain accountable for the messages they broadcast. In particular, it must be possible to uniquely determine the pseudonymous source of each signed message tuple (m, σ_i) and the certificate that authorises it ρ_i. Optionally, a suitable authority should be able to determine the canonical identity of a vehicle based on the (m, σ_i) and ρ_i that corresponds to some misbehaviour.

Corrupt CA Resistance The repercussions of certificate authority compromise should be minimised. The corruption of any single certificate authority must not allow the canonical identity of a vehicle to be determined from either the signed messages or the pseudonym certificates that are sent by the vehicle.

Sybil Attack Resistance Related to the abuse of pseudonyms by modified or compromised vehicles, a V2X scheme must limit the number of concurrent vehicle identities that can be used to sign messages. The optimal solution for road users without synchronised clocks is that Sybil attacks [73] are limited to 2 simultaneously valid pseudonyms per vehicle.

Revocation It is imperative that vehicles which misbehave by sending false information can be prevented from continued participation. It must be possible to remove a vehicle based on either the canonical identity or a valid, signed message (m, σ_i) and the corresponding pseudonym certificate ρ_i.

There are also a number of established performance requirements for V2X that we first

introduce in Section 2.5.5. In this chapter, we formulate the performance requirements for V2X as follows

Limited Connectivity A V2X scheme should support vehicles which have limited bandwidth and that suffer from intermittent connectivity. Bandwidth requirements should be minimised and V2X schemes should function securely despite limited or no access to online infrastructure.

Limited Resources Vehicles have relatively limited computational and storage capabilities and V2X schemes should be developed accordingly. The performance benchmark of at least 1,000 signature verifications per second established by the PRESERVE project [214] indicates that even just regular ECDSA verification, with no revocation mechanism, will challenge or exceed the computational capabilities of first and second generation connected vehicles. In addition, vehicles need to sign at least 10 broadcast messages per second, have limited certificate storage and are price-sensitive to the requirement for secure cryptographic hardware.

ETSI Compliant In this work, we require that V2X schemes are compatible with the ETSI TS 102 731 ITS security architecture standard. The standard is developed with respect to the security, privacy and performance requirements for V2X which are well established in the literature, and is the leading candidate for interoperable V2X communication throughout Europe.

5.5 System Model

In this section we describe the standard system and threat model that we use to develop and evaluate the IFAL V2X scheme. To remain compatible with the ETSI standard for V2X, we use the corresponding PKI architecture that is first introduced in Section 2.5.6.

In particular, the ETSI PKI comprises one or more of both an Enrolment Authority (EA) and an Authorisation Authority (AA). The vehicles and roadside infrastructure that participate in V2X are collectively termed ITS-Stations (ITS-S) and each ITS-S comprises an On-Board Unit (OBU) and a Trusted Element (TE). We assume the presence of a suitable Root Certificate Authority (RCA) that authorises the EA and AA credentials and has its verification key installed on each vehicle OBU during manufacture.

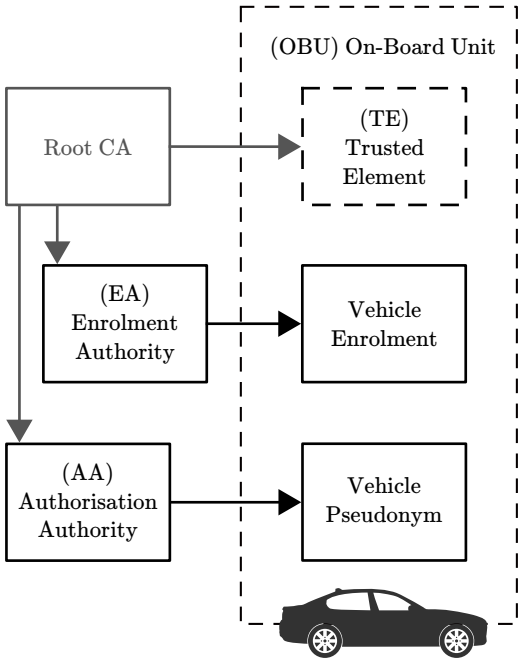


Figure 5.1: The ETSI PKI model.

Within the ETSI architecture, the EA is responsible for the provision and issuance of long-term ITS-S certificates and the role of the AA is to issue short-term pseudonym certificates that authorise an ITS-S to use a particular service. The role separation of the EA and AA functionality is a privacy mechanism that prevents any single certificate authority from linking vehicle pseudonym usage to a canonical identity.

5.5.1 Threat Model

In this chapter, we assume the following threat model which is based on the ETSI ITS standards.

For privacy, the threat model is limited by the requirement for accountability and the desire for law enforcement support that demands depseudonymisation. The use of role separation of the EA and the AA as a privacy mechanism means that we cannot preserve privacy when these authorities collaborate. As such we assume that the EA and the AA are honest-but-curious [110] adversaries, which do not collaborate, but that may opportunistically attempt to learn more than is necessary for their specified functionality. Concerning the role of the vehicle, a malicious or compromised vehicle OBU can always send arbitrary privacy-compromising information to an adversary. We assume that the vehicle OBU is an honest device and that the corresponding vehicle TE is a trusted and suitably audited secure hardware element that can generate an ECDSA public key pair and securely store the private key. All of this is directly inherited from the ETSI ITS standards.

For security, whilst the EA and the AA must still be honest-but-curious adversaries, the vehicle OBU is allowed to be malicious or compromised. This is realistic as the vehicle owner may try to modify their vehicle, or the vehicle may be infected with malware, and this must not break the authenticity or accountability of messages in a V2X scheme. The certificate authorities must always be able to revoke vehicles and the scheme must remain Sybil attack resistant, despite vehicle compromise.

5.6 Preliminaries

This section specifies the cryptographic primitives used by IFAL and defines the terms policy file and certificate file that are used throughout.

In IFAL the elliptic curve C is defined, in accordance with the ETSI standard [85], to be either NIST Curve P-256 [70] or BrainpoolP256r1 [163]. Both curves specify a base point G of prime order n, such that n is 256 bits long. The elliptic curve domain parameters C, G and n are global public values that we do not explicitly input to the

algorithms that utilise them.

IFAL requires the following cryptographic primitives. A hash function, a digital signature scheme, a public-key encryption scheme, a symmetric-key encryption scheme and two Key Derivation Functions (KDFs). For compatibility with the ETSI ITS standards we define the hash function to be SHA-256 [193], the digital signature scheme to be ECDSA, the public-key encryption scheme to be Elliptic Curve Integrated Encryption Scheme (ECIES) [133] and the symmetric-key encryption scheme to be the AES Cipher-based Message Authentication Code (CMAC) pseudorandom function [223].

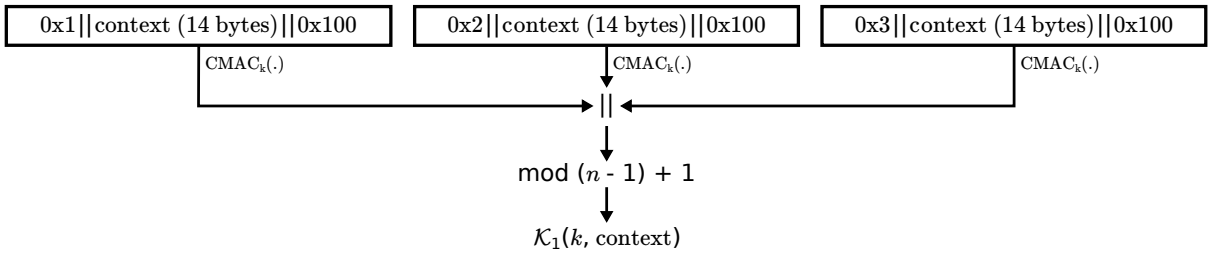


Figure 5.2: The NIST SP 800-108 KDF configuration we recommend for $\mathcal{K}_1$.

The two KDFs which we denote $\mathcal{K}_1$ and $\mathcal{K}_2$ require particular attention as their functionality is critical to the correct operation of IFAL. Both KDFs are to be constructed according to NIST SP 800-108 [222] with the following profiles:

- $\boldsymbol{\mathcal{K}_1}$ is used in the derivation of pseudonym public key values from some initial base point. In particular, $\mathcal{K}_1$ is required to provide non-zero integers modulo the prime curve order n. For $\mathcal{K}_1$ we first specify a NIST SP 800-108 KDF with an output size of at least $|n| + 64$ bits, for example the configuration shown in Figure 5.2, and then we restrict the output to non-zero elements modulo n. Where k is some master key and "context" provides the randomness for a particular KDF output, we use a technique proposed by the German Bundesamt für Sicherheit in der Informationstechnik (BSI)

[35] for which the KDF output is subject to the following additional operation:

$$\mathcal{K}_1(k, \text{context}) = \mathcal{K}(k, \text{context}) \mod (n-1) + 1$$

Using this technique, $\mathcal{K}_1$ is guaranteed to output the non-zero integers modulo n that are needed to derive pseudonym public key values in IFAL. As in the BSI standard, we assume that the deviation from a uniform distribution which results from this operation is too small to be exploited by any computationally-bound adversary.

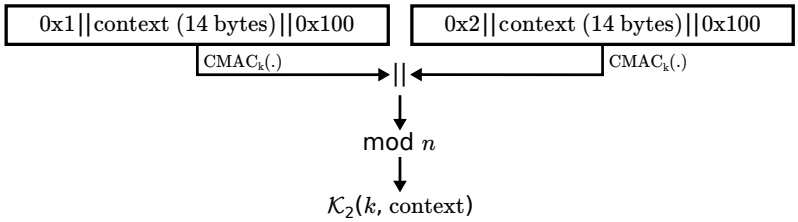


Figure 5.3: The NIST SP 800-108 KDF configuration we recommend for $\mathcal{K}_2$.

- $\boldsymbol{\mathcal{K}_2}$ is used to derive ECDSA instance keys and is part of a mechanism for providing optional law enforcement support in IFAL. As in $\mathcal{K}_1$, $\mathcal{K}_2$ is also required to provide non-zero integers modulo the prime curve order n. However, unlike $\mathcal{K}_1$, we additionally require that $\mathcal{K}_2$ has a rather surprising "recovery" property. Specifically $\mathcal{K}_2$ has the NIST SP 800-108 KDF configuration shown in Figure 5.3, with an output size of 256 bits. Under these constraints, $\mathcal{K}$ is actually bijective and also behaves as a symmetric encryption function, the 14 byte context value used to generate a particular $\mathcal{K}$ output can be recovered given the master key k. Whilst $\mathcal{K}_2$ is required to provide non-zero integers modulo n, it is sufficient to simply take the output modulo n as follows:

$$\mathcal{K}_2(k, \text{context}) = \mathcal{K}(k, \text{context}) \mod (n)$$

By construction the context values for $\mathcal{K}_2$ in IFAL include a counter value. In the highly improbable event that the output of $\mathcal{K}_2$ is zero, the counter can simply be incremented and the next value of $\mathcal{K}_2$ used instead without further consequence.

ECDSA Signature Scheme

IFAL makes extensive use of the ECDSA signature scheme, both in it's canonical implementation and two modified forms. We first introduce the standard ECDSA signature scheme [70] and then briefly discuss the two variants we will construct and utilise in the remainder of this chapter. Where $H_n : \{0,1\}^* \to \{0,1\}^{|n|}$ is a hash function with an output size of $|n|$ bits and O is the identity element of the group $\langle G \rangle$, the ECDSA signature scheme comprises the following triple of algorithms:

- **G** : On input the security parameter $1^{|n|}$ the algorithm selects the secret key $d \leftarrow \mathbb{Z}_n^*$ and computes the public key $P = d \times G$. The output is the public key pair (P, d).

- **S** : On input the secret key d and the message m, the algorithm selects the instance key $k \leftarrow \mathbb{Z}_n^*$ and computes the curve point $R = (x_1, y_1) = k \times G$, the random element $r = x_1 \bmod q$ and the signature element $s = k^{-1} \cdot (H_n(m) + d \cdot r) \bmod n$. The output is the ECDSA signature tuple $\tau = (r, s)$

- **V** : On input the public key P, the message m and the signature tuple $\tau = (r, s)$ the algorithm computes the following:

$$w = s^{-1} \bmod n$$
$$u_1 = H_n(m) \cdot w \bmod n$$
$$u_2 = r \cdot w \bmod n$$
$$R' = (x_1, y_1) = u_1 \times G + u_2 \times P$$

  If $P \in \langle G \rangle$, $P \times n = O$, $P, R' \neq O$, $r, s \in \mathbb{Z}_q^*$ and $r \equiv x_1 \bmod q$ then the output is **true** (accept) otherwise it is **false** (reject).

It is important to note by some convention [70], long-term ECDSA private keys are denoted with a d and ephemeral per-signature randomisation keys are denoted with a k. In this thesis we use the term "instance key" to refer to any such per-signature ECDSA randomisation key k.

In IFAL, the standard ECDSA algorithm described above is made use of extensively. All entities use the standard key-generation algorithm G to generate their public key pairs (d, P) and the standard verification algorithm V to verify all signatures. The standard signing algorithm S is used by the RCA to sign the long-term credentials of the EA and the AA, and also by the EA when signing a vehicle's long-term credential. IFAL also makes use of the following two variants of the ECDSA signing algorithm:

1. Deterministic ECDSA. In IFAL we make use of a deterministic ECDSA signing algorithm similar to the method described in RFC 6979 [190]. Rather than generating the instance key k uniformly at random for each new signature as in the canonical algorithm, we instead use the KDF $\mathcal{K}_2$ introduced above to derive instance keys using a secret known only by the AA, a counter value that never repeats and a pseudonymous vehicle identifier. This optional mechanism, when used by the AA to sign pseudonym certificates for a vehicle, permits the selective depseudonymisation of misbehaving vehicles in IFAL. This special ECDSA signing algorithm is used in Algorithm 2 in Section 5.7.2.

2. Computing the message digest on a separate device. In IFAL, when vehicles sign messages the work is divided between the OBU and the TE. In particular, the OBU computes the message digest and then the TE computes the remainder of the standard ECDSA signing algorithm. In Section 5.7.4 and Algorithms 3 and 4, we introduce some additional details and implement a splitting of the sign algorithm to allow the derivation of IFAL pseudonyms to occur on the vehicle OBU only. This allows the TE to remain oblivious to the IFAL pseudonym key derivation process,

enabling a single low-cost TE implementation to support both IFAL and the regular ETSI-standard architecture.

Policy Files

In IFAL, each vehicle requires a policy file that specifies the parameters of the pseudonym certificates that will be requested from the AA. As shown in Figure 5.4, an IFAL policy file specifies the following parameters:

1. T_{start}, the start time of the first certificate.
2. T_{valid}, the validity period of each certificate.
3. T_{overlap}, the time during which consecutive certificates are allowed to overlap in their validity period.
4. N_{certs}, the total number of certificates.
5. N_{epochs}, the total number of epochs.

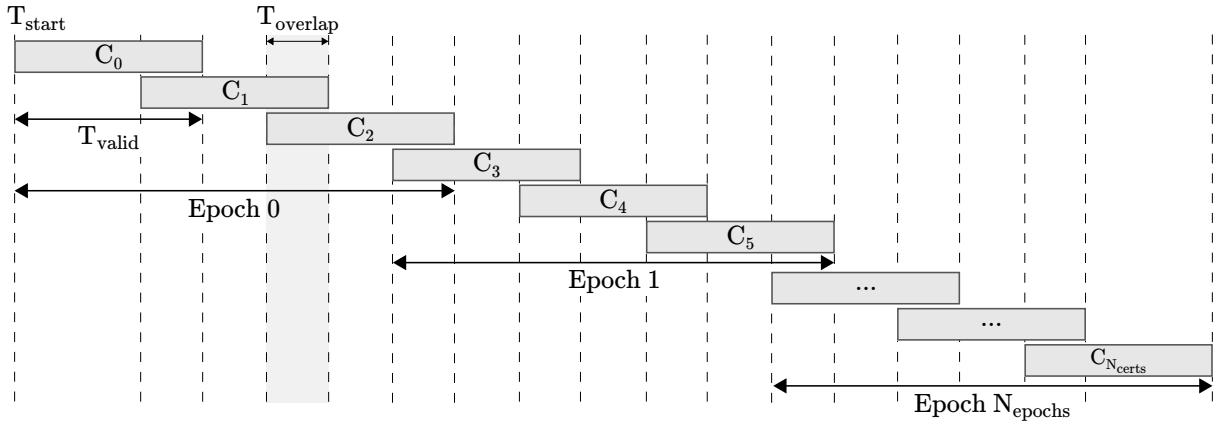


Figure 5.4: The IFAL policy file parameters.

To relax the requirement for precise clock synchronisation between different vehicles, the policy file specifies an overlap period T_{overlap} during which consecutive pseudonym certificates are simultaneously valid. Each vehicle is expected to change pseudonym after the minimum certificate validity period which is derived by subtracting the overlap period

from the validity period:

$$T_{\text{minimum}} = T_{\text{valid}} - T_{\text{overlap}}$$

Certificate Files

Certificate files contain a lifetime supply of inactive pseudonym certificates and are issued to vehicles in accordance with a policy file. In addition to the pseudonym certificates, each certificate file also comprises a digest of the corresponding policy file and a symmetric transport key k_T. The certificates in the file are divided evenly into epochs according to the vehicle policy. Each activation code in IFAL provides an epoch key, encrypted using the transport key k_T, which enables a vehicle to use the certificates in one particular epoch.

5.7 The IFAL Scheme

This section presents the full design and specification of our IFAL V2X scheme. For simplicity and without loss of generality, we only consider the most basic setting in which there is a single EA and just one AA. Furthermore we only consider the most essential ITS service, cooperative awareness. The IFAL scheme straightforwardly scales to an arbitrary number of ITS services and a correspondingly larger ecosystem of certificate authorities.

The differentiating approach of IFAL is to pre-issue vehicles with a lifetime supply of inactive pseudonyms that are only usable upon receiving small, time-delayed activation codes. In contrast to the ETSI standard, our scheme improves the trade-off between bandwidth, trust and privacy. Vehicles are provided with a certificate file that contains a unique, non-repeating pseudonym for every 5 minutes in a vehicles lifetime. The file is divided into epoch periods and an activation code is required to use the pseudonyms in any epoch. We introduce a novel key-diversification mechanism in the public-key setting that both eliminates the trade-off between pseudonym duration and bandwidth and reduces

the onboard certificate storage requirements in contrast to traditional V2X certificates. The related activation codes are small 128 bit encrypted keys and can easily be sent using any low-bandwidth insecure channel such as Short Message Service (SMS) messages. Potentially, IFAL can be deployed using existing infrastructure and does not depend on the availability of high-bandwidth cellular technologies in vehicles.

The IFAL scheme comprises five protocols: setup, initialisation, activation, usage and revocation. Briefly, setup first initialises the parameters of the scheme. Next, each vehicle runs the initialisation protocol in which they receive a long-term enrolment credential and a uid from the EA. Each enrolled vehicle sends its long-term credential and uid to the AA, which provides a certificate file containing a lifetime supply of inactive pseudonyms. Pseudonym certificates are activated periodically using activation codes that are sent to vehicles by the AA using the activation protocol. To sign messages, vehicles that have the appropriate activation code run the usage protocol which allows them to securely compute valid ECDSA signatures with respect to a particular pseudonym. The revocation protocol allows vehicles to be removed from the scheme in one of two ways, either using the canonical vehicle identity or using a signed message that corresponds to some vehicle misbehaviour.

5.7.1 Setup

Before vehicles can enrol in IFAL and request pseudonym certificates, the EA and the AA must first initialise the scheme parameters. Specifically, the EA generates the public key pair $(d_{\mathrm{EA}}, P_{\mathrm{EA}})$ and the AA generates the public key pair $(d_{\mathrm{AA}}, P_{\mathrm{AA}})$, the instance master key k_s and also initialises the secure counter value sc to zero.

For context, the EA will use the public key pair $(d_{\mathrm{EA}}, P_{\mathrm{EA}})$ to sign each vehicle's long-term enrolment credential. Similarly, the AA will use the public key pair $(d_{\mathrm{AA}}, P_{\mathrm{AA}})$ to sign each pseudonym certificate in each vehicle's certificate file. In addition, the AA will

use both the instance master key k_s and the secure counter sc to derive the randomising instance key for each pseudonym certificate signature. The instance master key k_s will enable each vehicle's uid to be securely embedded in each pseudonym certificate and the secure counter sc will ensure that no two signatures are ever signed using the same instance key, avoiding a well-known and critical ECDSA attack [245].

5.7.2 Initialisation Protocol

The initialisation protocol, shown in Figure 5.5, is run once for each new vehicle that joins the IFAL scheme. Vehicles receive a long-term enrolment credential from the EA and a certificate file of pseudonyms for cooperative awareness from the AA. In practice, the initialisation protocol is well suited to take place during manufacture of the vehicle and furthermore can be divided such that a vehicle can request pseudonyms for new services without necessarily requiring a new enrolment credential for each.

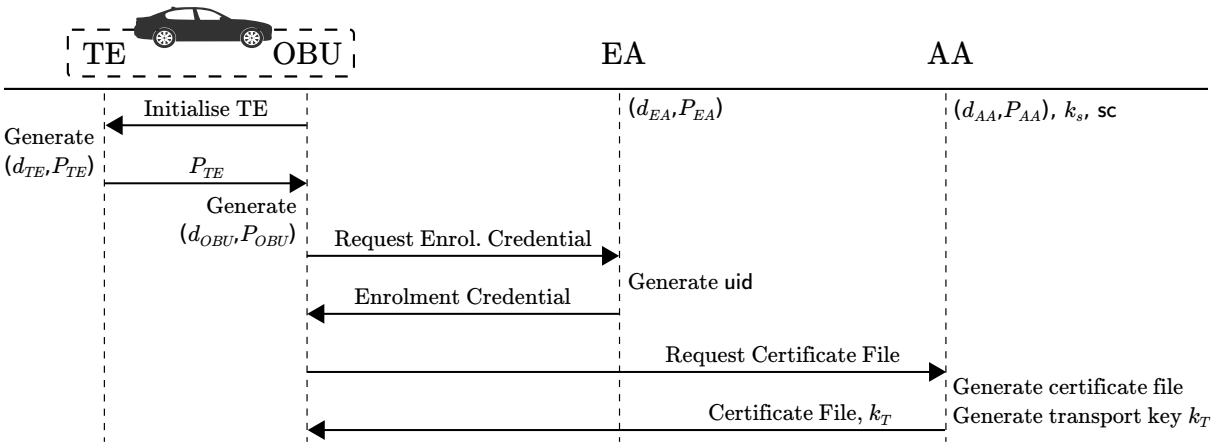


Figure 5.5: The IFAL initialisation protocol.

To begin with, the vehicle OBU has a policy file and a root certificate that were initialised during manufacture and an activation-code channel specification that is configured depending on the context of vehicle usage. Anticipated channel specifications include 4G LTE and SMS and are discussed in more detail in Section 5.7.3. The EA has the public key pair $(d_{\mathrm{EA}}, P_{\mathrm{EA}})$ and the AA has the public key pair $(d_{\mathrm{AA}}, P_{\mathrm{AA}})$, the instance master

key k_s and the secure counter value sc. The IFAL initialisation protocol is then as follows:

1. The OBU initialises the TE which generates the public key pair $(d_{\text{TE}}, P_{\text{TE}})$ and then sends the corresponding public key P_{TE} back to the OBU. The OBU generates its own public key pair $(d_{\text{OBU}}, P_{\text{OBU}})$ and then composes an enrolment request for the EA that includes the TE public key P_{TE}, the OBU public key P_{OBU}, the policy file and an activation-code channel specification. The vehicle OBU sends the enrolment request to the EA.
2. The EA has the public key pair $(d_{\text{EA}}, P_{\text{EA}})$ and processes enrolment requests that are received from vehicles. For each request, the EA checks that the vehicle is authorised to receive the certificate file corresponding to the policy in the request, generates a new uid and then creates and signs an enrolment credential on the newly created uid. The EA stores a record that links the vehicle uid with the enrolment request then sends the enrolment credential to the vehicle OBU.
3. The vehicle requests the certificate file from the AA by sending the enrolment credential, the TE public key P_{TE} and the corresponding policy file.
4. The AA has the public key pair $(d_{\text{AA}}, P_{\text{AA}})$, the symmetric instance master key k_s and the secure counter value sc. The AA validates the enrolment credential with respect to the EA public key P_{EA} and then creates a certificate file according to the vehicle policy. Specifically, the AA executes the `GenCertFile` algorithm which generates the certificate file for the vehicle and creates the set of symmetric epoch keys $(k_0, \ldots, k_{N_{\text{epochs}}-1})$, one for each epoch as specified in the policy file. The AA also creates a symmetric transport key k_T and stores a record that links the vehicle uid with the policy file, k_T and the set of epoch keys. Finally, the AA sends the certificate file and the transport key k_T to the vehicle.

Certificate File Creation

In the IFAL initialisation protocol just described, the AA issues each authorised vehicle with a certificate file that contains a lifetime supply of inactive pseudonym certificates. In more detail, the AA creates the certificate file and the associated epoch keys using the `GenCertFile` algorithm and signs each pseudonym certificate within the file using the `CertSign` algorithm.

Algorithm 1: `GenCertFile`
Run by : Authorisation Authority (AA)

private store : d_{AA}, k_s, sc
new inputs : policy file $= (T_{\text{start}}, T_{\text{valid}}, T_{\text{overlap}}, N_{\text{certs}}, N_{\text{epochs}}, \ldots)$, uid, P_{TE}

1 Create a new record for vehicle uid.
`/* Create epoch keys, later encrypted and termed activation codes. */`
2 **for** $j \leftarrow 0$ **to** $N_{epochs} - 1$ **do**
3 $\quad k_j \xleftarrow{\$} \{0,1\}^n$
4 Add epoch keys $k_0, \ldots, k_{N_{\text{epochs}}-1}$ to the record for uid.
`/* Create transport key used to encrypt epoch keys. */`
5 $k_T \xleftarrow{\$} \{0,1\}^n$
6 Create a new certificate file.
7 Derive metadata from policy file and write to certificate file.
8 Write transport key k_T to certificate file.
`/* Create and sign pseudonynm certificates. */`
9 **for** $i \leftarrow 0$ **to** $N_{certs} - 1$ **do**
10 $\quad j = i/N_{\text{epochs}}$
$\quad$ `/* Derive certificate validity from policy file. */`
11 $\quad$ start_validity $= T_{\text{start}} + i \cdot (T_{\text{valid}} - T_{\text{overlap}})$
12 $\quad$ end_validity $=$ start_validity $+ T_{\text{valid}}$
$\quad$ `/* Derive pseudonym public key. */`
13 $\quad P_i = \mathcal{K}_1(k_j, i) \times P_{\text{TE}}$
14 $\quad$ content $=$ start_validity $\|$ end_validity $\| P_i$
15 $\quad$ signature $=$ `CertSign(`content, uid`)`
16 $\quad$ certificate $=$ content $\|$ signature
17 $\quad$ Write certificate to certificate file.
18 **return** *file*

The `GenCertFile` algorithm takes as input the AA private key d_{AA}, the instance

master key k_s and the secure counter sc from the AAs private store. In addition, the algorithm also takes as input the policy file, uid and TE public key P_{TE} of the vehicle requesting the certificate file. The vehicle policy file, detailed in Section 5.6, specifies the start time of the first certificate T_{start}, the validity period of each certificate T_{valid}, the validity overlap period between consecutive certificates T_{overlap}, the total number of pseudonym certificates N_{certs} that should be in the file and the number of epochs N_{epochs} into which the certificates should be evenly divided.

The most important part of the `GenCertFile` algorithm is on Line 13, where each pseudonym public key $P_i = \mathcal{K}_1(k_j, i) \times P_{\text{TE}}$ is derived from the epoch key k_j, the certificate index i and the TE public key P_{TE}. This construction is equivalent to key-diversification in the public-key setting such that the master key is the TE private key d_{TE} that is stored on secure hardware. The signing key that corresponds to each pseudonym public key P_i is $k_i = d_{\text{TE}} \cdot \mathcal{K}_1(k_j, i)$ and can therefore only be derived by a vehicle that has both the correct TE private key d_{TE} and epoch key k_j. Epoch keys are withheld by the AA that issues the certificate file until they are required, thus providing a mechanism for passive revocation. In particular the epoch keys are encrypted, whereupon they are termed activation codes.

Each pseudonym in the certificate file is signed using the `CertSign` algorithm which takes as input the AA private key d_{AA}, the instance master key k_s, the secure counter sc and the vehicle uid. The `CertSign` algorithm is a variant of the deterministic ECDSA signature algorithm [190] in which the instance key k, which is a random bitstring in regular ECDSA, is derived from k_s, sc and the vehicle uid. In particular, the instance key k is derived using the invertible KDF $\mathcal{K}_2$ such that $k = \mathcal{K}_2(k_s, \mathsf{sc} \parallel \mathsf{uid})$.

The secure counter value sc is incremented for each signature and is used to ensure that no two certificates are signed using the same instance key, as this would expose the AA signing key d_{AA} [245]. When the secure counter reaches it's maximum value $2^{|sc|} - 1$ the AA must generate new parameters as shown in lines 2-6. Primarily, the AA must generate

Algorithm 2: `CertSign`
Run by : Authorisation Authority (AA)

private store : d_{AA}, k_s, sc
new inputs : content, uid
`/* Increment the secure counter. */`
1 $sc = sc + 1$
`/* If the secure counter limit is reached, re-key the AA */`
2 **if** $sc = (2^{|sc|} - 1)$ **then**
3 $\quad d_{\text{AA}} \xleftarrow{\$} \{0,1\}^n$
4 $\quad P_{\text{AA}} = d_{\text{AA}} \times G$
5 $\quad k_s \xleftarrow{\$} \{0,1\}^n$
6 $\quad sc = 0$
`/* Generate the ECDSA instance key deterministically then complete the signature using the standard algorithm. */`
7 $k = \mathcal{K}_2(k_s, \mathsf{sc} \parallel \mathsf{uid})$;
8 **if** $k = 0$ **then**
9 $\quad$ goto line 1
10 $R = (x_1, y_1) = k \times G$
11 $r = x_1 \; mod \; n$
12 $h =$`Hash(`content`)`
13 $s = (k^{-1}) \cdot (h + d_{\text{AA}} \cdot r) \; mod \; n$
14 **if** $r = 0$ **or** $s = 0$ **then**
15 $\quad$ goto line 1
16 **return** (r, s)

a new instance master key k_s and re-initialise the secure counter sc to zero. Since $2^{|sc|} - 1$ is an extremely large value this is also a convenient time to generate a new public key pair $(d_{\text{AA}}, P_{\text{AA}})$, ensuring that the computational upper bound on the number of signatures that can be requested from a signing oracle is not breached [110]. Lastly, the new AA public key P_{AA} must be signed by the RCA before pseudonyms signed using the new key will be considered valid by vehicles.

The derived instance signing key k allows for the message-based revocation of misbehaving vehicles and facilitates optional law enforcement support. The AA can use the instance master key k_s to recover the vehicle uid from the pseudonym certificate that authorises a V2X message. Law enforcement can compel the AA and the EA to collaborate so that the canonical identity of the vehicle can be determined.

5.7.3 Activation Protocol

The IFAL activation protocol is a periodic process in which trusted vehicles are supplied with activation codes. Each activation code is an encrypted epoch key that enables a vehicle to use the pseudonym certificates which correspond to a particular epoch in a certificate file. As shown in Figure 5.6, the activation protocol takes place between the AA, the EA and the vehicle OBU. The AA maintains a record for each vehicle uid that specifies the policy file, epoch keys $k_0, \ldots, k_{N_{\text{epochs}}-1}$ and the transport key k_T that was issued during the initialisation protocol.

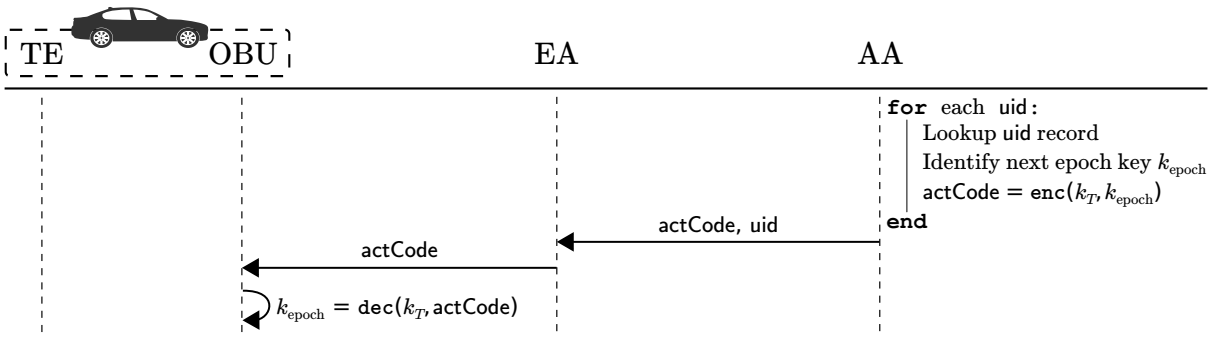


Figure 5.6: The IFAL activation protocol.

For each non-revoked vehicle uid, the activation protocol is as follows:

1. The AA uses the policy file associated with the vehicle uid to determine the next epoch key $k_{\text{epoch}} \in (k_0, \ldots, k_{N_{\text{epochs}}-1})$ that is required and also identifies the transport key k_T that will be used to encrypt it. The AA creates the activation code actCode = $\texttt{enc}(k_T, k_{\text{epoch}})$ and then sends the code, along with the vehicle uid, to the EA.
2. The EA has a record that links the vehicle uid with the activation-code channel specification that was agreed during the initialisation protocol. The EA sends the activation code actCode using the agreed channel.
3. The vehicle decrypts the activation code actCode using the transport key k_T to determine the epoch key $k_{\text{epoch}} = \texttt{dec}(k_T, \text{actCode})$.

Activation-Code Channel Specification

In the IFAL activation protocol, the EA distributes activation codes to vehicles on behalf of the AA. This separation of responsibilities is a privacy mechanism that ensures that no single authority can link the pseudonymous V2X messages sent by a vehicle to the corresponding canonical identity. The EA distributes activation codes using the activation-code channel specification that was agreed during the initialisation protocol. Since activation codes are small, 128 bit encrypted epoch keys they can be distributed over a wide range of channels. A key feature of IFAL is that the scheme supports vehicles with limited and even no connectivity. In the most extreme case, activation codes can be manually entered during vehicle servicing.

Since activation codes are secure encryptions of epoch keys, the channel used to transmit them does not require confidentiality. This approach keeps activation codes as small as possible, and is therefore suited to the widest range of vehicle connectivities, but is vulnerable to an active adversary that modifies or imitates the transmission of codes between the EA and the vehicle. When the activation-code channel is not authenticated (e.g. using TLS) then our activation protocol is vulnerable to an adversary that sends fake or corrupted activation codes to a vehicle. Such an adversary can cause a denial of service to the target vehicle which will spend time decrypting the activation codes and deriving the incorrect epoch and pseudonym private key. While this could be a problem for certain channels, the impact is limited to a denial of service of the target vehicle and can easily be avoided by using an authenticated-encryption mechanism to encrypt either the epoch keys or the activation-code channel. A target vehicle would still be able to receive and verify signatures from its peers and only its message sending capabilities would be denied.

One appealing possibility for distributing activation codes is using mobile SMS messages. In Europe, all new vehicles sold since March 2018 are legally required to be fitted with the "eCall" emergency call system [200]. eCall equips each new vehicle with a mobile Subscriber

Identity Module (SIM) card that in the event of an accident, if not overridden by the driver, automatically calls the emergency services and provides the location of the vehicle. eCall means that all new European vehicles are already fitted with the communication equipment that is necessary for widespread activation code distribution.

5.7.4 Usage Protocol

The IFAL usage protocol is run each time that a vehicle signs a message. The protocol is based on the standard ECDSA signing algorithm, however the message digest is subject to an additional transformation process and the steps of the algorithm are shared between the vehicle TE and OBU. Specifically, the vehicle OBU and TE jointly sign a message using the `MessageSign` and `IFALSign` algorithms, respectively.

Algorithm 3: `MessageSign`
Run by : Vehicle On-Board Unit (OBU)

private store : policy file $= (T_{\text{start}}, T_{\text{valid}}, T_{\text{overlap}}, N_{\text{certs}}, \ldots)$
new inputs : m, t
`/* Determine the certificate index i */`
1 $i = (t - T_{\text{start}})/(T_{\text{valid}} - T_{\text{overlap}})$
`/* Determine the epoch key index */`
2 epoch $= i/N_{\text{certs}}$
3 If no k_{epoch} return error.
4 $k_i = \mathcal{K}_1(k_{\text{epoch}}, i)$
5 $h =$ `Hash(`m`)`
6 $h' = h \cdot k_i^{-1}\ mod\ n$
7 $(r, s) =$ `IFALSign(`h'`)`
8 $s' = s \cdot k_i\ mod\ n$
9 **return** (r, s')

To sign the message m, the vehicle OBU executes the `MessageSign` algorithm which takes as input m, the current time t and the policy file. The `MessageSign` algorithm determines the current pseudonym index i and epoch key k_{epoch} using t and the policy file which includes the start time of the first certificate T_{start}, the minimum validity period of each pseudonym $T_{\text{valid}} - T_{\text{overlap}}$ and the total number of pseudonym certificates N_{certs}. Using the pseudonym index i and the epoch key k_{epoch}, the OBU derives the pseudonym

private key $k_i = \mathcal{K}_1(k_{\text{epoch}}, i)$ that corresponds to the public key $P_i = \mathcal{K}_1(k_j, i) \times P_{\text{TE}}$ that is derived by the AA when creating the certificate file using the `GenCertFile` algorithm.

Once the pseudonym private key k_i has been derived, the remainder of the `MessageSign` algorithm proceeds as follows. Firstly, the OBU computes the digest $h = \texttt{Hash}(m)$ of the message and then transforms h using the modular inverse of the pseudonym private key k_i^{-1}. The transformation of the message digest is similar to Chaum's blind signatures [47] and allows the TE to remain oblivious to the IFAL pseudonym key derivation process. This means that a single, low-cost TE implementation can support both IFAL and the regular ETSI-standard architecture. The TE codebase is hence kept to an absolute minimum, just the standard ECDSA algorithm without taking the message digest, enabling efficient auditing and the use of a standard smartcard device. These steps contribute towards meeting the requirements for limited resources and ETSI compliance introduced in Section 5.4.

Algorithm 4: `IFALSign`
Run by : Vehicle Trusted Element (TE)

private store : d_{TE}
new inputs : h'

1 $k \xleftarrow{\$} \text{Z}_n \setminus \{0\}$
2 $R = (x_1, y_1) = k \times G$
3 $r = x_1 \; mod \; n$
4 $s = k^{-1} \cdot (h' + d_{\text{TE}} \cdot r) \; mod \; n$
5 **if** $r = 0 \; OR \; s = 0$ **then**
6 | goto line 1
7 **return** (r, s)

The transformed message digest $h' = h \cdot k_i^{-1} \; mod \; n$ is sent to the vehicle TE which has the TE private key and executes the `IFALSign` algorithm. The algorithm is simply the standard ECDSA signing algorithm [70], without the hash function, and returns the ECDSA signature (r, s) on the transformed message digest h'. The TE sends the ECDSA signature (r, s) to the vehicle OBU.

Finally, to complete the `MessageSign` algorithm, the OBU unblinds the signature by multiplying the s-component by the pseudonym private key k_i. The resulting signature tuple (r, s') is a valid signature on the message m with respect to the pseudonym public key P_i.

Signature Correctness

The `MessageSign` algorithm computes ECDSA signatures that are valid with respect to the pseudonym public keys which are derived by the `GenCertFile` algorithm. Specifically, where k_j is the epoch key and P_{TE} is the TE public key, the ith pseudonym public key in the certificate file is computed $P_i = \mathcal{K}_1(k_j, i) \times P_{\text{TE}}$ and the corresponding private key is $d_{\text{i}} = d_{\text{TE}} \cdot \mathcal{K}_1(k_j, i)$.

Here we show that the signature tuple (r, s') is a valid signature with respect to the pseudonym public key P_i. Recall that $h' = h \cdot k_i$, $r = x_1 \ mod \ n$, $s = k^{-1} \cdot (h' + d_{\text{TE}} \cdot r) \ mod \ n$ and $s' = s \cdot k_i \ mod \ n$, then s' can also be written:

$$
\begin{aligned}
s' &= k_i \cdot k^{-1} \cdot (h' + d_{\text{TE}} \cdot r) \ mod \ n \\
\therefore \ s' &= k_i \cdot k^{-1} \cdot (h \cdot k_i^{-1} + d_{\text{TE}} \cdot r) \ mod \ n \\
\therefore \ s' &= k^{-1} \cdot (h + k_i \cdot d_{\text{TE}} \cdot r) \ mod \ n \\
\therefore \ s' &= k^{-1} \cdot (h + d_i \cdot r) \ mod \ n
\end{aligned}
$$

Hence the signature tuple (r, s') is a valid ECDSA signature with respect to the pseudonym public key $P_i = \mathcal{K}_1(k_j, i) \times P_{\text{TE}} = k_i \times P_{\text{TE}} = k_i \cdot d_{\text{TE}} \times G = d_i \times G$.

5.7.5 Revocation Protocol

In IFAL there are two different mechanisms for revocation and correspondingly, two different protocols. IFAL supports both identity-based and message-based vehicle revocation with the following two protocols:

Identity-Based Revocation The first revocation protocol is based on the EA being provided with the canonical identity of the vehicle that should be removed from the scheme. This could occur when the vehicle is taken off the road by its owner, or after the vehicle is "written off" by the insurer following an accident. The EA uses the canonical vehicle identity to look up the uid that was issued during the initialisation protocol. The EA sends the uid to the AA who then removes the associated certificate file record. During the activation protocol the AA will no longer issue new epoch keys to the revoked vehicle and so it will be removed from the scheme after, at most, one epoch.

Message-Based Revocation The second revocation protocol is based on the AA being provided, ideally by a suitable authority and with accompanying evidence, with pseudonym certificates that correspond to a vehicle that has misbehaved. For example, the vehicle may have been modified to broadcast misleading information or may have been involved in a hit-and-run accident. The signature on the pseudonym certificate will be an ECDSA signature tuple (r, s) that was output by the `CertSign` algorithm during the initialisation protocol. The AA has the instance master key k_s and can recover the vehicle uid from the pseudonym certificate. Specifically, the signature value $s = k^{-1} \cdot (h + d_{\mathrm{AA}} \cdot r) \mod n$ can be rearranged to provide the instance key $k = s^{-1} \cdot (h + d_{AA} \cdot r) \mod n$ that was used to create the signature. In the `CertSign` algorithm, the instance key k is deterministically produced using the invertible KDF $\mathcal{K}_2$ such that $k = \mathcal{K}_2(k_s, \mathsf{sc} \parallel \mathsf{uid})$. Correspondingly, the uid of the vehicle that the pseudonym certificate was issued to can be recovered using the inverse KDF as follows, $\mathcal{K}_2^{-1}(k_s, k) = \mathsf{sc} \parallel \mathsf{uid}$. The AA removes the certificate file record associated with the uid and optionally, collaborates with the EA to also resolve the canonical vehicle identity. Once the uid record is removed, the vehicle will be revoked within one epoch period or less.

5.8 V2X Formal Model

This section describes our formalisation of the ETSI system model in terms of the standard security and privacy requirements for V2X. In particular, we define a V2X scheme and then formalise the terms "secure V2X" and "privacy conscious V2X".

5.8.1 Preliminaries

Here we introduce the standard syntax and security definitions for digital signatures and Key Derivation Functions (KDFs) that are the foundations of our security definitions. Most of it is standard and we refer the reader to Goldreich [110] and Krawczyk [153], respectively, for a more thorough treatment.

Digital Signature Scheme

A digital signature scheme Σ is a triple $(\mathcal{G}, \mathcal{S}, \mathcal{V})$ of efficient algorithms that satisfy the following two conditions:

1. $\mathcal{G}$ is a key generation algorithm that, on input the security parameter 1^η, outputs the pair of bitstrings (k, P).
2. For every pair (k, P) output by $\mathcal{G}(1^\eta)$ and for every message $m \in \{0,1\}^*$, the signing algorithm $\mathcal{S}$ and the verification algorithm $\mathcal{V}$ satisfy:

$$Pr[V(P, m, S(k, m)) = 1] = 1$$

The standard formal security definition for digital signature schemes is Existential Unforgeability under Chosen-Message Attack (EUF-CMA) [111]. The definition refers to an experiment that is played between an efficient adversary $\mathcal{A}$ and an oracle $\mathcal{O}_\mathcal{S}$ that will sign arbitrary messages. The public-key signature scheme experiment is defined as follows.

EUF-CMA$_{\mathcal{O}_\mathcal{S}}(1^\eta, \mathcal{A})$:

1. The oracle $\mathcal{O}_\mathcal{S}$ simulates the key-generation algorithm G which generates the key pair (k, P) and then provides the adversary $\mathcal{A}$ with the target verification key $P^\star$.
2. Challenge: Polynomially many times, the adversary $\mathcal{A}$ submits a message m to the oracle $\mathcal{O}_\mathcal{S}$ and learns the corresponding signature $\sigma = \mathcal{O}_\mathcal{S}(k, m)$.
3. Output: The adversary $\mathcal{A}$ outputs the pair of bitstrings $(m^\star, \sigma^\star)$.

The adversary $\mathcal{A}$ is deemed successful and wins the EUF-CMA experiment if and only if the following two conditions hold:

1. The message $m^\star$ is different from all queries made by the $\mathcal{A}$ to the signing oracle $\mathcal{O}_\mathcal{S}$. In other words, $m^\star$ is different from any string in $\mathcal{O}_\mathcal{S}(k, m)$.
2. The tuple $(m^\star, \sigma^\star)$ corresponds to a valid message-signature pair relative to the verification key $P^\star$ and therefore $\mathcal{V}(P^\star, m^\star, \sigma^\star) = 1$.

Definition 5.8.1 (Secure Signature Scheme) A public-key signature scheme $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ is said to be secure if for all efficient adversaries $\mathcal{A}$, the probability of $\mathcal{A}$ winning the experiment **EUF-CMA**$_{\mathcal{O}_\mathcal{S}}(1^\eta, \mathcal{A})$ is a negligible function of the security parameter η.

Key Derivation Function

A KDF is a function that is used to generate cryptographically strong pseudorandom keys from some inadequate initial source of keying material. Specifically, a KDF is an algorithm $\mathcal{K}$ that takes as input a bitstring k and a length parameter l. Optionally, a salt value r and a context variable x are also input. The algorithm outputs a bitstring of length l bits.

The security of a KDF depends on the input bitstring k which is sampled from a source of keying material ϕ. Formally, ϕ is an efficient algorithm that takes as input the security parameter 1^η and outputs the probability distribution tuple (k, α). In the tuple output by

the source ϕ, k is the bitstring that will be input to the KDF and α is auxiliary knowledge about k which is known to the adversary. For example, in a Diffie-Hellman application [68] the bitstring k will have the value g^{xy} and the auxiliary knowledge α would comprise the group parameters (p, q, g) and the public key values (g^x, g^y).

The standard security definition for a KDF demands that the output bitstring is indistinguishable from a random bitstring of the same length. The definition refers to an experiment that is played between an adversary $\mathcal{B}$ and an oracle $\mathcal{O}_\mathcal{K}$ that will derive keys for adaptively chosen context and length queries. The secure KDF experiment is defined as follows:

(t,q,ε)-Secure-KDF$_{\mathcal{O}_\mathcal{K}}(\eta, q, \mathcal{B})$

1. The oracle $\mathcal{O}_\mathcal{K}$ simulates the source algorithm ϕ which generates the probability distribution tuple (k, α) and then provides the adversary $\mathcal{B}$ with the auxiliary knowledge α.
2. For $i = 1, \ldots, q' \leqslant q$, $\mathcal{B}$ adaptively submits chosen context and length queries to the key derivation oracle $\mathcal{O}_\mathcal{K}$ and learns the corresponding KDF output $k'_i = \mathcal{O}_\mathcal{K}(k, x_i, l_i)$.
3. Challenge: The adversary $\mathcal{B}$ chooses a context and length query (x, l) such that $x \notin \{x_1, \ldots, x_{q'}\}$. In other words the context x has not previously been submitted to $\mathcal{O}_\mathcal{K}$.
4. A bit $b \in \{0, 1\}$ is chosen at random. If $b = 0$ then $\mathcal{O}_\mathcal{K}$ provides $\mathcal{B}$ with the KDF output $k' = \mathcal{O}_\mathcal{K}(k, x, l)$, otherwise $\mathcal{O}_\mathcal{K}$ provides a random bitstring $k' = \{0, 1\}^l$ of length l bits.
5. $\mathcal{B}$ may repeat Step 2, subject to the total number of queries remaining less than q and the context not being equal to the challenge context x.
6. The adversary $\mathcal{B}$ outputs a bit $b' \in \{0, 1\}$. $\mathcal{B}$ wins the game if $b' = b$.

Definition 5.8.2 (Secure KDF) A KDF $\mathcal{K}$ is said to be (t, q, ε)-secure with respect to a source of keying material ϕ, if for all efficient adversaries $\mathcal{B}$ that run in time t and make

at most q queries, the probability of $\mathcal{B}$ winning the **(t,q,ε)-Secure-KDF** experiment is less than $1/2 + \varepsilon$.

5.8.2 V2X Scheme

Definition 5.8.3 (V2X Scheme) A V2X scheme Π comprises the following quartet of efficient algorithms and protocols:

- An algorithm `CreatePKI` that outputs the public and private PKI parameters $(\mathsf{PP}, \mathsf{SP})$ which are sets that include the public and private signing keys belonging to each RCA, EA and AA, respectively. In addition, the public PKI parameters PP also includes the certificates ρ_{EA} and ρ_{AA} that authorise the EA and the AA, respectively, in relation to the RCA public key P_{RCA}.
- An algorithm `CreateVehicle` which outputs the public key pairs of a vehicle OBU $(d_{\text{OBU}}, P_{\text{OBU}})$ and the associated TE $(d_{\text{TE}}, P_{\text{TE}})$.
- An interactive protocol `EnrolVehicle` that is run between the EA and a vehicle.
- An interactive protocol `AuthoriseVehicle` that is run between the AA and a vehicle.

5.8.3 Secure V2X

The main security requirement for a V2X scheme is message authentication. In this section, we capture this requirement by formalising the term "secure V2X" scheme in relation to the authentication experiment **Auth-V2X**. To improve the completeness of our definition, we overload the standard digital signature verification algorithm $\mathcal{V}$ from Section 5.8.1 as follows. In the authentication experiment **Auth-V2X**, the verification algorithm $\mathcal{V}$ takes as input the full certificate chain that authorises the verification key P. Specifically, $\mathcal{V}$ takes as input the V2X scheme root certificate public key P_{RCA}, the AA certificate ρ_{AA}, the pseudonym certificate ρ, the message m and the signature σ. The modified verification algorithm $\mathcal{V}$ returns `true` if and only if:

1. The tuple (m, σ) is a valid digital signature with respect to the definition of a secure signature scheme in Section 5.8.1. In other words, where P is the verification key in the pseudonym certificate ρ, $\mathcal{V}(P, m, \sigma)$.
2. There is a valid certificate chain from the pseudonym certificate ρ to the root certificate public key P_{RCA}. For example, $\mathcal{V}(P_{\text{RCA}}, \rho_{\text{AA}}, \sigma_{\rho_{\text{AA}}}) = 1$ and $\mathcal{V}(P_{\rho_{\text{AA}}}, \rho, \sigma_{\rho}) = 1$.

The authentication experiment is played between an adversary $\mathcal{C}$ and a V2X scheme oracle $\mathcal{O}_{\Pi}$ that simulates a number of vehicles. The experiment is as follows:

Auth-V2X$_{\mathcal{O}_{\Pi}}(1^{\eta}, \mathcal{C})$

---

1. The V2X scheme oracle $\mathcal{O}_{\Pi}$ takes as input the security parameter 1^{η} and simulates the `CreatePKI` algorithm which outputs the public and private PKI parameters $(\mathsf{PP}, \mathsf{SP})$ and also the `CreateVehicle` algorithm which creates N_V vehicles with identities $V_1, \ldots, V_{N_V}$. The oracle $\mathcal{O}_{\Pi}$ provides the adversary $\mathcal{C}$ with the public parameters $\mathsf{PP} = (P_{\text{RCA}}, P_{\text{EA}}, \rho_{\text{EA}}, P_{\text{AA}}, \rho_{\text{AA}})$.
2. Challenge: For an arbitrary polynomial duration of $\mathcal{C}$'s choosing, $\mathcal{O}_{\Pi}$ provides $\mathcal{C}$ with the message-signature-certificate triples (m, σ, ρ) that are sent by the vehicles $V_1, \ldots, V_{N_V}$. Afterwards, the adversary $\mathcal{C}$ outputs the triple of bitstrings $(m^{\star}, \sigma^{\star}, \rho^{\star})$.

The adversary $\mathcal{C}$ is deemed successful and wins the **Auth-V2X** experiment if:

1. The message $m^{\star}$ is different from any m provided by the V2X scheme oracle $\mathcal{O}_{\Pi}$.
2. The triple $(m^{\star}, \sigma^{\star}, \rho^{\star})$ corresponds to a valid message-signature pair $(m^{\star}, \sigma^{\star})$ with respect to the pseudonym certificate $\rho^{\star}$ and the certificate chain from ρ to P_{RCA} is valid. In other words, $\mathcal{V}(P_{\text{RCA}}, \rho_{\text{AA}}, \rho^{\star}, m^{\star}, \sigma^{\star}) = 1$.

Definition 5.8.4 (Secure V2X) We say that a V2X scheme Π is secure if for all efficient adversaries $\mathcal{C}$, the probability of $\mathcal{C}$ winning the **Auth-V2X** experiment is a negligible function of η.

5.8.4 Privacy Conscious V2X

The main privacy requirements for V2X are unlinkability and pseudonymity. In this section, we formalise the term "privacy conscious V2X" which captures these requirements in relation the privacy experiment **t-Priv-V2X**. We recall from Section 3.2.2 that the scope of achievable privacy in V2X is limited to considering intermediate adversaries that only have a temporally intermittent, or spatially partial observational capacity. The periodic location information provided by broadcast messages in V2X is inherently susceptible to techniques such as MHT [202] that are reliably able to track and identify individual sources [117, 259].

The main technique for providing privacy in V2X is with pseudonym certificates. Pseudonyms allow a vehicle to send messages without revealing its identity but also ensure that the vehicle remains accountable. It is only when vehicles change pseudonym at a time or place that the adversary is unable to witness that there is any chance of preventing long-term tracking. In particular, it has been shown that even "perfectly unlinkable" broadcast messages that use a different pseudonym for each message are still vulnerable to long-term identification and tracking [160, 22].

In this work we separate ourselves from the issue of messages that provide inherently identifiable location information and focus instead on the cryptographic qualities of a V2X scheme that might undermine vehicle privacy. This permits us to quantify the privacy leakage of the V2X scheme in a way that is independent from human behaviour or vendor-specific implementation details. We define privacy conscientiousness for V2X in terms of the cryptographic linkability of message signatures and certificates, disentangled from the situationally dependent message contents. In particular, we do not allow the contents of a broadcast message to contribute to the adversaries advantage.

We capture the cryptographic linkability of a V2X scheme with the following experiment that is played between an adversary $\mathcal{D}$ and the "obscured" V2X scheme oracle $\mathcal{O}_{\not\Pi}$. The

obscured oracle $\mathcal{O}_{\text{Ч}}$ is like the V2X oracle $\mathcal{O}_{\Pi}$ from the **Auth-V2X** experiment in Section 5.8.3. However, rather than providing the adversary with the messages that are broadcast by vehicles, $\mathcal{O}_{\text{Ч}}$ sends messages that are chosen uniformly at random from a message distribution $\mathcal{M}$. Here we use the notion of a "vehicle reference" analogously to how pointers are used in computer programming languages and our experiment is similar to the off-line RFID privacy model developed by Garcia et al. [106]. The **t-Priv-V2X** experiment is as follows:

t-Priv-V2X$_{\mathcal{O}_{\text{Ч}}}(1^{\eta}, \mathcal{D})$

---

1. The obscured V2X scheme oracle $\mathcal{O}_{\text{Ч}}$ takes as input the security parameter 1^{η} and simulates the `CreatePKI` algorithm which outputs the public and private PKI parameters $(\mathsf{PP}, \mathsf{SP})$ and also the `CreateVehicle` algorithm which creates N_V vehicles with references $V_1, \ldots, V_{N_V}$. The oracle $\mathcal{O}_{\text{Ч}}$ provides the adversary $\mathcal{D}$ with the public parameters $\mathsf{PP} = (P_{\text{RCA}}, P_{\text{EA}}, \rho_{\text{EA}}, P_{\text{AA}}, \rho_{\text{AA}})$ and the vehicle references $V_1, \ldots, V_{N_V}$.
2. Challenge: After an arbitrary polynomial duration of $\mathcal{D}$'s choosing, during which $\mathcal{O}_{\text{Ч}}$ provides $\mathcal{D}$ with all of the obscured message-signature-pseudonym triples (m, σ, ρ) that are sent by the vehicles $V_1, \ldots, V_{N_V}$, $\mathcal{D}$ chooses a target vehicle reference $V^{\star} \in \{V_1, \ldots, V_{N_V}\}$.
3. The oracle $\mathcal{O}_{\text{Ч}}$ invalidates all of the original vehicle references $V_1, \ldots, V_{N_V}$, chooses a bit $b \in \{0, 1\}$ at random and then pauses for the polynomial duration t. During the time t, no messages from any of the vehicles $V_1, \ldots, V_{N_V}$ are sent to the adversary $\mathcal{D}$. After time t and if $b = 0$ then $\mathcal{O}_{\text{Ч}}$ resumes simulating only the vehicle that had the reference $V^{\star}$, otherwise $\mathcal{O}_{\text{Ч}}$ simulates a different vehicle chosen uniformly at random from $\{V_1, \ldots, V_{N_V}\} \setminus V^{\star}$.
4. After an arbitrary polynomial duration of $\mathcal{D}$'s choosing, during which $\mathcal{O}_{\text{Ч}}$ provides $\mathcal{D}$ with all of the obscured message-signature-pseudonym triples (m, σ, ρ) that are sent by the remaining vehicle, $\mathcal{D}$ outputs the bit $b' \in \{0, 1\}$.

The adversary $\mathcal{D}$ is deemed successful and wins the **t-Priv-V2X** experiment if $b' = b$.

Definition 5.8.5 (Privacy Conscious V2X) A V2X scheme Π is said to be privacy conscious if for all efficient adversaries $\mathcal{D}$, the probability of $\mathcal{D}$ winning the **t-Priv-V2X** experiment is less than $1/2 + \varepsilon$.

5.9 The Security and Privacy of IFAL

In this section we show that IFAL is a secure and privacy conscious V2X scheme.

5.9.1 IFAL is a Secure V2X Scheme

This section shows that IFAL is a secure V2X scheme with respect to Definition 5.8.4. In particular, we reduce the security of IFAL to the unforgeability of the underlying signature scheme.

Theorem 5.9.1 Let Σ be a secure signature scheme with respect to Definition 5.8.1, then IFAL is a secure V2X scheme with respect to Definition 5.8.4.

Proof. Let us assume for contradiction that IFAL is not a secure signature scheme. In relation to the **Auth-V2X** experiment this means that there is an adversary $\mathcal{C}$ who after being provided with the public parameters $\mathsf{PP} = (P_{\text{RCA}}, P_{\text{EA}}, \rho_{\text{EA}}, P_{\text{AA}}, \rho_{\text{AA}})$ and all of the message-signature-certificate triples (m, σ, ρ) which are sent by N_V vehicles during an arbitrary period of observation, manages to craft a triple $(m^\star, \sigma^\star, \rho^\star)$ such that $m^\star$ is unique from any m sent by any vehicle and $\mathcal{V}(P_{\text{RCA}}, \rho_{\text{AA}}, \rho^\star, m^\star, \sigma^\star) = 1$.

We show how to use such an adversary $\mathcal{C}$ to break the security of the underlying signature scheme. Specifically, we construct an adversary $\mathcal{A}$ that uses $\mathcal{C}$ to win the **EUF-CMA** experiment. $\mathcal{A}$ simulates the full IFAL PKI environment including the initialisation, activation and usage protocols as specified in Section 5.7. $\mathcal{A}$ generates the public and private PKI parameters $(\mathsf{PP}, \mathsf{SP})$. The public parameters $\mathsf{PP} = (P_{\text{RCA}}, P_{\text{EA}}, \rho_{\text{EA}}, P_{\text{AA}}, \rho_{\text{AA}})$

include the RCA public key and the public keys and certificates of the EA and the AA. The private parameters $\mathsf{SP} = (d_{\mathrm{RCA}}, d_{\mathrm{EA}}, d_{\mathrm{AA}}, k_s, \mathsf{sc})$ include the private keys of the RCA and the EA as well as the private key, instance master key and secure counter of the AA.

To begin, the adversary $\mathcal{A}$ simulates N_V vehicles by first creating the corresponding OBU and TE public key pairs $(d_{\mathrm{OBU}}, P_{\mathrm{OBU}})$ and $(d_{\mathrm{TE}}, P_{\mathrm{TE}})$, respectively. For each vehicle, $\mathcal{A}$ simulates runs the initialisation and activation protocols so that the certificate file is generated and the pseudonym keys are available. $\mathcal{A}$ chooses the target pseudonym validity period $e^\star$ and the target vehicle $V^\star \in \{V_1, \ldots, V_{N_V}\}$. When emulating $V^\star$ during the target epoch $e^\star$, rather than using the vehicle private keys $(d_{\mathrm{OBU}}, d_{\mathrm{TE}})$ and simulating the usage protocol, $\mathcal{A}$ will instead use the signing oracle $\mathcal{O}_{\mathcal{S}}$ from the **EUF-CMA** game. For all other vehicles $\mathcal{A}$ will simulate the usage protocol as usual.

After an arbitrary period of adversary $\mathcal{C}$'s choosing, during which $\mathcal{A}$ will simulate periodic message sending and provide $\mathcal{C}$ with the message-signature-certificate tuples (m, σ, ρ) sent by all N_V vehicles, $\mathcal{C}$ will terminate and output a triple of bitstrings $(m^\star, \sigma^\star, \rho^\star)$. By hypothesis, $m^\star$ is unique from any m sent by any vehicle and $\mathcal{V}(P_{\mathrm{RCA}}, \rho_{\mathrm{AA}}, \rho^\star, m^\star, \sigma^\star) = 1$. In other words, $(m^\star, \sigma^\star)$ is a valid message-signature tuple and there is a valid certificate chain from the root public key P_{RCA} to the pseudonym certificate $\rho^\star$.

Since IFAL is based on the ETSI V2X standards, the signed message $(m^\star, \sigma^\star)$ must be an IEEE WAVE based CAM crafted according to the ETSI CAM security profile [88]. In particular, $(m^\star, \sigma^\star)$ is a `IEEE1609dot2 SignedData` structure as shown in Figure 5.7 such that $m^\star$ is the triple (`hashID`, `tbsData`, `signer`) and $\sigma^\star$ is the `signature` element.

In order to win the **EUF-CMA** experiment, the adversary $\mathcal{A}$ needs to output a signed message $(m^\star, \sigma^\star)$ such that:

1. The message $m^\star$ is different from all of the queries made by $\mathcal{A}$ to the oracle $\mathcal{O}_{\mathcal{S}}$.
2. The tuple $(m^\star, \sigma^\star)$ is a valid message-signature pair relative to the verification key $P^\star$ and therefore $\mathcal{V}(P^\star, m^\star, \sigma^\star)$.

```
SignedData ::= SEQUENCE {
  hashId HashAlgorithm,
  tbsData ToBeSignedData,
  signer SignerIdentifier,
  signature Signature
}
```

Figure 5.7: The `SignedData` specification from the IEEE 1609.2 standard [131].

```
SignerIdentifier ::= CHOICE {
  digest  HashedId8,
  certificate SequenceOfCertificate,
  self  NULL,
  ...
}
```

Figure 5.8: The `SignerIdentifier` specification from the IEEE 1609.2 standard [131].

Condition 1. holds because $m^\star$ was not queried to the oracle $\mathcal{O}_\mathcal{S}$. As the target vehicle $V^\star$ is chosen randomly by $\mathcal{A}$, the probability that C attacks $V^\star$ is $1/N_V$. Condition 2. holds because $(m^\star, \sigma^\star)$ is a valid message-signature tuple with respect to the certificate chain from P_{RCA} to $\rho^\star$. With respect to the CAM message structure in Figure 5.7, this also means that if the `SignerIdentifier` element shown in Figure 5.8 is a `digest` then the adversary $\mathcal{A}$ has previously received the corresponding certificate $\rho^\star$.

The advantage of the adversary $\mathcal{C}$ in winning the **Auth-V2X** experiment is therefore the probability that $\mathcal{C}$ attacks vehicle $V^\star$ multiplied by the advantage of $\mathcal{A}$ against the signature scheme. Since $\mathcal{C}$ may attack either the signature on the message during the certificate validity period $e^\star$, or the signature at any stage of the certification path, the advantage of $\mathcal{C}$ is further divided by the certification path length ℓ and the total number of pseudonym validity periods N_E over which $\mathcal{A}$ provides tuples (m, σ, ρ) to $\mathcal{C}$.

$$\mathsf{Adv}_{\mathcal{C}}^{\textbf{Auth-V2X}} = \frac{\mathsf{Adv}_{\mathcal{A}}^{\textbf{EUF-CMA}}(1^\eta)}{\ell \cdot N_V \cdot N_E}$$

5.9.2 IFAL is a Privacy Conscious V2X Scheme

This section shows that IFAL is a privacy conscious V2X scheme with respect to Definition 5.8.5. We reduce the privacy of IFAL to the security of the underlying KDF scheme.

Informally, an adversary cannot link vehicle pseudonyms to a single source because all pseudonym public keys are the product of a random vehicle TE public key and an output

from the secure KDF $\mathcal{K}_1$. To win the **t-Priv-V2X** experiment with a non-negligible probability, an adversary must be able to learn or know something that can differentiate pseudonyms sent from one vehicle from those sent by another. Since a secure KDF has the property that the output is indistinguishable from a random bitstring of the same length, and random bitstrings do not portend anything about future random bitstrings, no adversary can link different pseudonyms to a single source.

IFAL is a privacy conscious V2X scheme because a secure KDF is used to derive the pseudonym public key values that are issued to vehicles. In more detail, each pseudonym public key P_i is computed as the product of the vehicle TE public key P_{TE} and the output of the KDF $\mathcal{K}_1$ which is seeded with the epoch key k_j and the context i, $P_i = \mathcal{K}_1(k_j, i) \times P_{\text{TE}}$. Correspondingly, vehicles use an implicit KDF to derive the pseudonym private key that is required to sign messages that are valid with respect to P_i. We define the implicit pseudonym private key KDF as follows:

$$\mathcal{K}_{\text{pseudo}}(k_j, i) = \mathcal{K}_1(k_i, i) \cdot d_{\text{TE}} \mod n$$

Theorem 5.9.2 If $\mathcal{K}_1$ is a secure KDF with respect to Definition 5.8.2 then $\mathcal{K}_{\text{pseudo}}$ is also a secure KDF.

By definition, the output of KDF $\mathcal{K}_1$ is indistinguishable from a random bitstring in the field $\mathbb{Z}_n^*$. Since the vehicle TE private key d_{TE} is a generated securely by tamper-resistant hardware and modular multiplication over a prime modulus n is uniformly distributed in $\mathbb{Z}_n^*$, $\mathcal{K}_{\text{pseudo}}$ is a secure KDF. Even if d_{TE} is generated non-uniformly, provided that it is in the finite field $\mathbb{Z}_n^*$ then $\mathcal{K}_{\text{pseudo}}$ is still a secure KDF.

Theorem 5.9.3 If $\mathcal{K}_{\text{pseudo}}$ is a secure KDF then IFAL is a privacy conscious V2X scheme with respect to Definition 5.8.5.

From Theorem 5.9.2 it follows that every pseudonym private key d_{pseudo} that is used to

sign CAM in IFAL is a random bitstring in the field $\mathbb{Z}_n^*$. Each corresponding pseudonym public key is computed by multiplying d_{pseudo} by the elliptic curve base point G. In other words, $P_i = d_{\text{pseudo}} \times G = d_i \cdot d_{\text{TE}} \times G = \mathcal{K}_1(k_j, i) \cdot d_{\text{TE}} \times G$. Multiplying a random bitstring by an elliptic curve point does not yield a secure KDF because, on all standard curves, a curve point is highly-distinguishable from a random bitstring [16]. Instead, for our notion of a privacy conscious V2X, we analogously formulate that the pseudonym public key must be indistinguishable from a random point on the same curve. It therefore suffices that the pseudonym private key d_{pseudo} is output by a secure KDF.

Proof. Let us assume for contradiction that IFAL is not a privacy conscious V2X scheme. This means that there is an adversary $\mathcal{D}$ that manages with a non-negligible probability to win the **t-Priv-V2X** experiment. We build an adversary $\mathcal{B}$ that uses $\mathcal{D}$ to win the **(q,t,ε)-Secure-KDF** experiment and therefore breaks the security of the underlying KDF.

To begin, the adversary $\mathcal{B}$ simulates the full IFAL PKI environment including the protocols, algorithms and roles specified in Section 5.7. $\mathcal{B}$ simulates N_V vehicles by generating the appropriate public key pairs and then simulating the initialisation and activation protocols. $\mathcal{B}$ chooses a target epoch $e^\star$ and a target vehicle $V^\star$. For each vehicle $\mathcal{B}$ emulates message sending by periodically simulating the usage protocol to generate message-signature-pseudonym triples (m, σ, ρ). When simulating $V^\star$ during epoch $e^\star$, rather than using the KDF $\mathcal{K}_1$ to generate the pseudonym private key, $\mathcal{B}$ uses the key derivation oracle $\mathcal{O}_\mathcal{K}$ from the **(q,t,ε)-Secure-KDF** experiment. As in the **t-Priv-V2X** experiment, each vehicle message m is chosen uniformly at random from a message distribution $\mathcal{M}$. Adversary $\mathcal{B}$ provides $\mathcal{D}$ with a reference to each vehicle $V_1, \ldots, V_{N_V}$ and provides all of the vehicle broadcast message triples (m, σ, ρ).

After an arbitrary polynomial duration, adversary $\mathcal{D}$ will provide $\mathcal{B}$ with a target vehicle reference V_i. $\mathcal{B}$ will invalidate all of the vehicle references $V_1, \ldots, V_{N_V}$, choose a

bit $b \in \{0, 1\}$ and then pauses for the duration t. If $b = 0$ then $\mathcal{B}$ resumes simulating vehicle V_i only, otherwise $\mathcal{B}$ resumes simulating a different vehicle chosen at random from $\{V_1, \ldots, V_{N_V}\} \setminus V_i$. Again $\mathcal{B}$ provides $\mathcal{D}$ with all of the message-signature-pseudonym triples (m, σ, ρ) that are sent by the remaining vehicle. Eventually $\mathcal{D}$ will terminate and output a bit $b' \in \{0, 1\}$.

By hypothesis, $b' = b$ with a probability significantly higher than $1/2 + \varepsilon$. This means that adversary $\mathcal{D}$ has succeeded in distinguishing the pseudonyms sent by vehicle V_i from those of another and that one of the following must hold true:

1. The silent period t is less than the minimum certificate validity period T_{minimum}. If $t \leqslant T_{\text{minimum}}$ then $V^\star$ will sign messages using a pseudonym certificate that has already been witnessed by $\mathcal{D}$. The adversary will be able to win the experiment with a probability of 1.
2. $\mathcal{D}$ found linkable information in the pseudonym certificates ρ that were sent by V_i.
3. The adversary $\mathcal{D}$ broke the security of the KDF $\mathcal{K}_1$ that generated V_i's pseudonym public keys and was able to distinguish the values from random points on the same curve. $\mathcal{D}$ must have chosen to attack the target vehicle so that $V_i = V^\star$ and during the target epoch $e^\star$.

Condition 1. only holds if $t <= T_{\text{minimum}}$ and Condition 2. does not hold provided that the pseudonym certificates are created in accordance with the ETSI standards and that they do not contain any linkable information. Condition 3. holds because, by hypothesis, $\mathcal{D}$ was able to output $b' = b$ with a probability significantly higher than $1/2 + \varepsilon$. This means that $\mathcal{D}$ was able to distinguish the pseudonym key values sent by V_i from those of any other vehicle $\{V_1, \ldots, V_{N_V}\} \setminus V_i$.

Where N_{periods} is the integer number of epoch periods that the experiment is run over and provided that $t > T_{\text{minimum}}$, the advantage of the adversary $\mathcal{D}$ in winning the **t-Priv-V2X** experiment is the probability that $\mathcal{D}$ attacks the vehicle $\mathcal{V}^\star$ during the target

epoch $e^\star$ multiplied by the advantage of $\mathcal{B}$ against the KDF scheme.

$$\mathsf{Adv}^{\textbf{t-Priv-V2X}}_{\mathcal{D}} = \frac{\mathsf{Adv}^{\textbf{(q,t,}\varepsilon\textbf{)-Secure-KDF}}_{\mathcal{B}}(1^\eta)}{N_V \cdot N_{\text{periods}}}$$

5.10 Evaluation and Performance

In this section we argue that the IFAL scheme we have presented in Section 5.7 meets the security, privacy and performance requirements for V2X from Section 5.4.

The primary security requirement for V2X is that there is a mechanism for determining the authenticity and integrity of broadcast messages. We show in Section 5.9.1 that IFAL is a secure V2X scheme and provide a reduction from the security of IFAL to the security of the underlying signature scheme.

There are four main privacy requirements for V2X. IFAL achieves pseudonymity because the structure of broadcast CAM and the pseudonym certificates that authorise them do not reveal the canonical vehicle identity. Correspondingly, and as shown in Section 5.9.2 where we prove that IFAL is a privacy conscious V2X scheme, IFAL achieves cryptographic unlinkability and does not reveal any more information than is necessary for providing pseudonymous authentication. Accountability in IFAL is accomplished because each message is authenticated using a secure signature scheme and a pseudonym certificate that is signed by the AA. Messages undeniably originate from a source that not only has been provided with the epoch key that was necessary to produce the signature, but that also has the same vehicle TE that was involved in the certificate file issuance process. Finally, IFAL satisfies the requirement for corrupt certificate authority resistance because, as in the standard ETSI ITS architecture, the EA and the AA use role separation to ensure that no single authority can link pseudonymous broadcast messages to a canonical vehicle identity.

Concerning depseudonymisation and revocation, in IFAL each pseudonym certificate

is signed using the `CertSign` algorithm from Section 5.7.2. In particular, the instance signing key that is used to sign each pseudonym is computed using the invertible KDF $\mathcal{K}_2$ such that the uid of the vehicle can always be recovered by the AA. In Section 5.7.5 we show that IFAL supports both identity-based and message based revocation and that optionally, the AA can collaborate with the EA to provide law enforcement support. The ETSI standards [83] indicate that the AA should be implemented using an HSM which would execute the key generation, `GenCertFile` and `CertSign` algorithms. The certificate file and transport key would only leave the HSM once encrypted for the vehicle OBU and activation codes would also be generated within the device. To provide greater defence against collaborating certificate authorities, a misbehaviour authority could be established and tasked with operating a recovery HSM that provides instance key encryption and decryption for the AA.

V2X schemes are required to support vehicles with limited computational and storage resources. IFAL only introduces a small computational overhead when compared to the standard. In particular, the most time-critical operation in V2X is signature verification, requiring at least 1,000 operations per second. In IFAL, signature verification is unchanged from the ETSI standard and is just one regular ECDSA verification operation. For signing, which has a modest 10 per second performance requirement, IFAL adds just one KDF and one modular inverse operation per pseudonym, and two additional modular multiplications per message. These small overheads can easily be accommodated within existing V2X hardware without a significant performance impact. With regards to storage requirements, a typical 5 year supply of IFAL pseudonym certificates requires as little as 32.1 megabytes. We evaluate the IFAL certificate file creation and storage requirements more thoroughly in Section 5.10.2.

With regards to supporting vehicles with intermittent connectivity or limited bandwidth, IFAL has improved support over the standard. Vehicles are issued with a lifetime supply of

inactive pseudonym certificates that are divided into epochs. Each epoch comprises one or more pseudonym certificates and correspondingly, the activation codes that enable vehicles to sign messages in IFAL contain epoch keys that allow the private signing keys for one or more pseudonyms to be derived. The policy file that defines the number of pseudonyms per epoch and the duration of each certificate allows IFAL to support vehicles with a wide range of different connectivities. Activation codes are only 128 bits in size and, even when supplemented with an additional 40 bit epoch and certificate file identifier for convenience, can be represented as a 28 character alphanumeric string. One appealing possibility for distributing activation codes is using mobile SMS messages and leveraging the existing eCall system [200] that has been fitted to all new European vehicles since March 2018. Since activation codes are encrypted, they can be sent over an insecure channel and in the most extreme case of entirely unconnected vehicles, can even be entered manually during vehicle servicing.

V2X schemes are required to limit the impact of vehicles that abuse their pseudonymity. IFAL optimally limits Sybil attacks that abuse concurrently valid pseudonyms because, at most, each vehicle only has two concurrently valid pseudonym certificates. Each vehicle is issued a unique pseudonym for every certificate period in the certificate file and it is only during the small overlap period, which is necessary for harmonising vehicles without synchronised clocks, that two pseudonyms are simultaneously valid. In contrast to the U.S. DOT SCMS certificate pooling approach in which there are 20-40 concurrently valid pseudonyms changed weekly, IFAL has much better resistance to Sybil attacks.

Lastly, in this work we required that IFAL was compatible with the ETSI TS 102 731 ITS security architecture standard [85]. IFAL conforms to the standard, is developed around the same PKI model and uses the same cryptographic primitives, but also offers improved support for vehicles with limited and intermittent connectivity.

5.10.1 Bandwidth Profile

This section details the improved trade-off between bandwidth, privacy and revocation that IFAL represents. Recall that IFAL caters to early deployments of V2X in which there are vehicles with a range of different, limited connectivities and there is likely to be intermittent access to central resources. IFAL is superior to the ETSI standard ad-hoc issuance of time-limited pseudonym certificates because it is not necessary to compromise between the bandwidth required for sending each certificate and the privacy afforded by the validity period of each pseudonym.

The lower-bound on the amount of online bandwidth that IFAL saves is 896 bits per pseudonym certificate. This bound is established when considering a minimum ECDSA certificate size of 1024 bits and then subtracting the 128 bit size of an IFAL activation code. An absolute upper bound is not possible to establish without introducing additional constraints since, by design, an activation code enables a vehicle to derive ECDSA signatures for an arbitrary number of pseudonyms.

The number of pseudonyms corresponding to each activation code is defined by the epoch duration which is specified in the policy file of each vehicle. In their pre-standardisation survey of pseudonym change management [84], ETSI provides the example of using 5 minute pseudonym validity periods and a 1 minute overlap duration when adopting a fixed-time pseudonym change strategy. In Figure 5.9, we compare the bandwidth needed to implement the ETSI ad-hoc pseudonym issuance mechanism using this strategy with the bandwidth savings provided

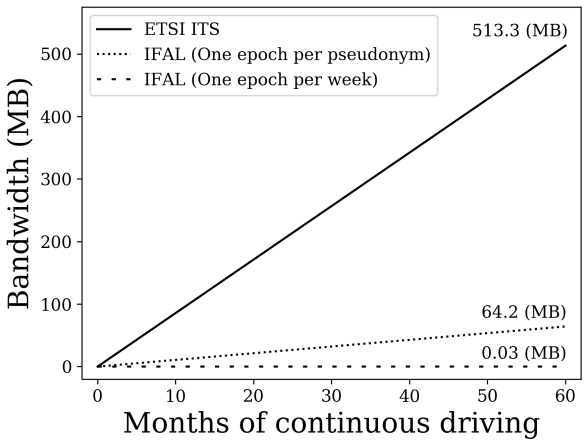


Figure 5.9: ETSI vs IFAL: Certificate issuance bandwidth required for a continuous supply of 5 minute pseudonym certificates.

by the IFAL. We consider both the lower-bound, in which IFAL reduces bandwidth by 896 bits per pseudonym yet maintains revocation granularity, as well as a more relaxed setting similar to the U.S. DOT recommendations for which we adopt an epoch duration of one week.

In more detail and using the ETSI suggested 5 minute pseudonym validity period and 1 minute overlap duration, ad-hoc certificate issuance requires 288 KB per day or less. This is simply the 1024 bit size of each certificate multiplied by the quantity, 288, needed for 24 hours of continuous driving. This can be considered an upper bound on the bandwidth required since highly-connected vehicles could request a smaller number that better fit the owner's normal driving patterns. A more developed analysis of the sort needed to establish average bandwidth requirements for ad-hoc certificate issuance would require expensive behavioural modelling and is outside the scope of this work.

IFAL introduces an additional epoch duration parameter which we evaluate in two different configurations. Firstly we consider the upper-bound that occurs when the epoch duration is equal to the pseudonym validity period. In this configuration, a vehicle requires one activation code for every pseudonym in the certificate file and so the total bandwidth requirement per day, 36 KB, is the 128 bit activation code size multiplied by the 288 pseudonyms needed every 24 hours. As in the ETSI case this is also an upper bound in the sense that highly-connected vehicles could request a smaller number of activation codes that better fit normal driving patterns.

Secondly, we consider the setting in which the ETSI epoch duration parameter is set to a period of one week. In this configuration an activation code allows a vehicle to derive the signing keys for each of the 2016 unique 5 minute pseudonym certificates which, in total, span a 7 day period. The total online bandwidth required is just 128 bits per week and can easily be communicated using a range of bandwidth mechanisms such as mobile SMS.

In the case of both the ETSI and IFAL certificate issuance mechanisms the only constraint on how often certificates and activation codes, respectively, must be sent to each vehicle is that it must be before the pseudonym is needed. When using ad-hoc certificate issuance there is a trade-off between the bandwidth that is available to each vehicle and the privacy, in terms of certificate validity period, that can be provided. Vehicles with limited or intermittent connectivity must rely on a smaller number of pseudonyms, each with a longer validity period. IFAL improves this situation by allowing a fixed size, 128 bit activation code to correspond to an arbitrary number pre-installed but otherwise unusable of certificates.

There is an additional trade-off with respect to bandwidth and privacy. The further in advance certificates are issued, or the longer the epoch duration, the greater the period before a vehicle can be revoked. All certificates or activation codes that have been issued must expire before passive revocation can take place. Finding the optimal balance between epoch duration, revocation and privacy is an interesting direction for future research.

5.10.2 Experimental Results

To evaluate the performance of IFAL, we have created a proof of concept reference implementation in C++ based on the Crypto++ library. Our reference implementation is open source and freely available at `https://github.com/hkscy/IFAL`.

Since signature verification on the vehicle is unchanged, namely a standard ECDSA verification operation, and IFAL does not add significant computational overhead to message signing, we focus on the performance of the server-side `GenCertFile` and `CertSign` algorithms which are executed by the AA. We wrote an IFAL policy specifying a certificate file with the following parameters:

- $N_{\text{certs}} = 525,600$
- $N_{\text{epochs}} = 20$
- $T_{\text{valid}} = 5$ minutes
- $T_{\text{overlap}} = 2$ minutes

In other words, the file will contain 5 years worth of consecutive pseudonym certificates, each valid for 5 minutes. An epoch will last 91 days and there are 20 total epochs in the file. Using a standard desktop computer we were able to compute the certificate file containing a 5 year supply of pseudonym certificates in 9.03 seconds on average. The certificate file comprises $525,600$ ECDSA certificates and therefore requires $525,600 \cdot 1,024 \approx 64.2$ MB of storage on the vehicle OBU. Optionally, if the OBU has the resources to derive the corresponding public key for each pseudonym, the certificate file can be halved in size to just 32.1 MB.

5.11 Chapter Summary

In this chapter we have presented the Issue First Activate Later (IFAL) scheme, a practical and secure improvement to the leading European security architecture candidate for ITS.

IFAL is based on a novel key diversification mechanism in the public-key setting that improves support for vehicles with limited and intermittent connectivity. Each vehicle is pre-issued with a certificate file that contains a lifetime supply of inactive pseudonym certificates. The file is divided into epochs and vehicles receive small, time-delayed activation codes that enable them to use the pseudonyms in a particular epoch. By adjusting the epoch duration, vehicles with a wide range of different connectivities can be provided with the same level of cryptographic unlinkability. Activation codes are small, 128 bit encrypted values that are particularly well suited to being sent as SMS messages, potentially avoiding the need for new infrastructure. IFAL supports the revocation and optional depseudonymisation of vehicles based on messages that correspond to misbehaviour. Revoked vehicles are passively denied the activation codes required for their future participation, therefore removing the need for CRL and offering improved verification latency over the previous proposals.

We have shown that IFAL meets the ETSI standard security and privacy requirements, is provably secure and privacy conscious in a formal setting and has favourable performance in our reference implementation. IFAL is efficient, standards compliant and a good candidate for mainstream deployment.

Chapter 6

Vehicular Direct Anonymous Attestation

In this chapter we look beyond the immediately practical improvements to the ETSI V2X standards developed in the last chapter, namely IFAL, and consider privacy enhancements that are suited to less-constrained V2X deployments. In particular, we address the threat of colluding certificate authorities with a new Vehicular DAA (VDAA) scheme which harmonises the standard requirements for V2X introduced in Section 2.5.4 with the strong privacy guarantees of Direct Anonymous Attestation (DAA). This work is aimed at future deployments of V2X which do not incur the same computational and bandwidth constraints which guided our previous work. Indeed, the activation tokens which are a key part of our IFAL solution are not needed for our VDAA scheme.

VDAA uniquely addresses, despite subverted and collaborating certificate authorities, the challenge of preventing long-term vehicle tracking in V2X whilst retaining centralised authority over vehicle revocation. Our scheme includes a novel construction that optimally limits Sybil attacks by restricting each vehicle to one pseudonym request per epoch. We also present a new security model for VDAA and show that we can reduce the unforgeability and unlinkability of our ECDSA broadcast messages to the security of the underlying DAA scheme.

6.1 Motivation

The leading standards for V2X [86, 29] both propose the use of ECDSA pseudonym certificates for providing privacy. In the standards, and elsewhere in the literature [23, 188, 102, 208, 256, 189], pseudonyms provide privacy for V2X in two dimensions. Firstly, pseudonyms are used to protect vehicles from the type of long-term tracking that intends to uniquely identify individual drivers. Vehicles change pseudonym multiple times per journey such that, when pseudonym change occurs outside of the observational reach of an adversary, the vehicle cannot be distinguished from any other vehicle that also changed pseudonym during the same period of observational unavailability. In the previous chapter we develop a new V2X architecture, IFAL, which ensures that each vehicle has a unique pseudonym for every 5 minute period in its lifetime and that the cryptographic signatures on broadcast messages do not provide linkability between different pseudonyms.

The second way in which pseudonyms are commonly used to improve privacy in V2X is to prevent any single certificate authority from linking pseudonymous vehicle signatures to a canonical vehicle identity. In both of the leading security standards for V2X, vehicles are issued with a long-term ECDSA enrolment certificate that is repeatedly used over a vehicles lifetime to request multiple short-term pseudonymous authorisation certificates. In IFAL too, a single pseudonymous uid is associated with the set of all pseudonyms and activation codes that are issued to each canonical vehicle. In all of these exclusively ECDSA-based systems, only role-separation between the long-term enrolment authorities and the short-term authorisation authorities ensures that no single compromised authority can link a pseudonymous vehicle signatures to a canonical identity.

Whilst the standards offer some privacy protection from honest-but-curious [110] certificate authorities, neither standard protects vehicles from certificate authorities which are dishonest or that collaborate. This limitation has also been recognised by the Data

Protection Working Party who have called for new technical measures that address the problem [183]. Developing these measures is challenging, in part, because of the standardisation of ECDSA for V2X. In both of the leading V2X standards, ECDSA is used for the internal certification of authorities, signing enrolment certificates, signing pseudonym certificates and for signing the broadcast messages that are sent by vehicles. Although ECDSA is well suited to V2X because it offers small signature sizes and low-latency message verification, it is less flexible then other schemes such as Schnorr [212] or Camenisch-Lysyanskaya [41] signatures. In particular, the inability to re-randomise an ECDSA signature [162] makes it impossible to strongly protect the privacy of vehicles that request pseudonyms by repeatedly presenting a long-term ECDSA certificate.

Several works have developed enhanced privacy techniques for V2X based on using a more flexible signature scheme [102, 50, 256, 8, 208]. From a privacy perspective, a particularly attractive candidate for use in V2X in Direct Anonymous Attestation (DAA). DAA is an anonymous group signature scheme first introduced by Brickell et al. [30] and since standardised by the Trusted Computing Group (TCG) who include it in their Trusted Platform Module (TPM) specification [143]. In DAA, group members comprise a TPM and a host that work together to sign messages with respect to a basename. Members receive a blind signature on their long-term credential and then authenticate as a group member by proving in zero knowledge [92] that they have such a signed credential. DAA offers strong privacy guarantees that include unforgeability, non-frameability and unlinkability even when the group issuer is corrupt [38]. These properties are desirable for V2X as they overlap with the standard requirements but also provide unlinkability despite a dishonest issuer. Whilst the strong privacy guarantees of DAA make it an attractive candidate for use in V2X, the computational costs, large signature sizes and the potential for abuse of anonymous credentials prohibit its straightforward application.

In this chapter we are motivated to reconcile the strong privacy guarantees of DAA,

including unlinkability despite a dishonest issuer, with the fast verification speed, small signature size and standards compliance of ECDSA signatures for V2X authentication.

6.2 Contributions

VDAA is a new V2X scheme that addresses the need to protect vehicles from subverted or collaborating certificate authorities. VDAA harmonises the strong privacy guarantees of DAA with the standard requirements for V2X introduced in Section 2.5.4. In contrast to the leading standards for V2X [88, 29], VDAA offers a stronger security model in which only the vehicle OBU must be trusted for privacy. Uniquely in the literature [102, 50, 256, 8, 208] our scheme retains centralised authority over vehicle revocation, efficient standards-compliant ECDSA signatures on broadcast messages and does not require the TPM to be trusted for privacy. The main contributions in this chapter are as follows:

1. We present our new Vehicular DAA (VDAA) scheme which harmonises the strong privacy guarantees of DAA with the standard requirements for V2X. In particular, we retain the ECDSA CAM signatures that are required by both of the leading V2X standards for their fast verification speed and small signature size.
2. We uniquely address the problem of preventing long-term vehicle tracking by dishonest, subverted and colluding certificate authorities whilst still retaining centralised vehicle revocation. In our security model, only the vehicle OBU needs to be trusted for vehicle privacy.
3. We introduce a novel construction that optimally limits Sybil attacks by restricting vehicles to a single ECDSA pseudonym credential per epoch. Vehicles that attempt to retrieve multiple pseudonyms for the same epoch are denied and forfeit unlinkability.
4. We model the VDAA system and formalise its unforgeability and unlinkability notions. We provide a reduction from the unforgeability and unlinkability of our

scheme to the security properties of the underlying DAA and ECDSA schemes.

The remainder of this chapter has the following structure. To begin, we summarise our notation and provide a brief reminder of the standard security and privacy requirements for V2X. In Section 6.6 we provide the new threat model that motivates the development of our scheme. In Section 6.8 we present the full details of our new VDAA scheme and in Section 6.9 we provide the corresponding formalisation and our game-based security definitions for unforgeability and unlinkability. In Section 6.10 we show that the security of VDAA can be reduced to the security of the underlying DAA and ECDSA schemes before finally, in Section 6.11, we evaluate VDAA with respect to the standard requirements.

6.3 Notation

In this chapter we use $x \leftarrow S$ to denote some x chosen uniformly at random from the set S. We let $|x|$ denote the bit size of x, let $x \parallel y$ express the concatenation of x and y and let $x \times G$ denote the scalar multiplication of point G by x. We distinguish between DAA and ECDSA public key pairs using the notation $(\mathsf{sk}, \mathsf{pk})$ and (d, P), respectively. In addition, we let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ denote groups of large prime order n and we let G, g_1, $\overline{g}$, $\tilde{g}$ and g_2 denote the generators such that $\mathbb{G}_1 = \langle G \rangle = \langle \overline{g} \rangle = \langle \tilde{g} \rangle = \langle g_1 \rangle$ and $\mathbb{G}_2 = \langle g_2 \rangle$. We let e be a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ such that:

- $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ is an efficiently computable homomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ with $\psi(g_1) = g_2$.
- $\forall x \in \mathbb{G}_1, y \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_n$, $e(x^a, y^b) = e(x, y)^{ab}$.
- e is non-degenerate, in other words $e(g_1, g_2) \neq 1$.

For denoting signature proofs of knowledge of discrete logarithms, and signature proofs of the validity of statements about discrete logarithms, we use the standard notation introduced by Camenisch and Stadler [42]. For example, $\mathsf{SPK}[\alpha, \beta : \overline{y} = \overline{g}^{\alpha} \wedge y_1 = g_1^{\beta}](m)$

denotes the "signature proof of knowledge" upon m and of integers α and β such that $\overline{y} = \overline{g}^{\alpha}$ and $y_1 = g_1^{\beta}$ holds. To distinguish between proofs with TPM contribution and those without we use SPK^* and SPK, respectively. We use the notation $\mathsf{NIZK}[(w) : \text{statement}(w)](\mathsf{ctxt})$ from [38] to denote any non-interactive zero-knowledge proof that is bound to a context ctxt and proves knowledge of a witness w such that statement(w) is true. To remain indifferent, where possible, to the underlying signature schemes used in our constructions we also use the notation PBSig and PBVf to denote generic partially-blind signing and verification algorithms respectively.

In the formal security setting we use the term efficient to mean solvable using a probabilistic polynomial-time Turing machine with an error probability of less than 1/2.

6.4 Requirements

We have already introduced the standard security, privacy and trust requirements for V2X. In Chapter 2 we identify the core requirements from the literature and standards and in Chapter 5 we reformulate the same standard requirements in a more precise and symbolic setting.

Briefly, the core security and privacy requirements for V2X PKI that are converged upon in the main standards [87, 29] and the literature [256, 258, 50, 102] are authentication, pseudonymity, unlinkability, accountability, corrupt CA resistance, Sybil attack resistance and revocation. Rather than to restate them here, we refer the reader to Section 5.4 from the previous chapter for a thorough definition of each of these requirements.

6.5 System Model

In this chapter we retain the standard ETSI PKI model [85] for V2X which we introduce in Section 5.5. The ETSI model comprises one or more vehicles, Enrolment Authorities (EA) and Authorisation Authorities (AA). There is also a RCA although we leave this implicit

to simplify our discussion. For simplicity and without loss of generality we consider a single EA, just one AA and an implicit single Root Certificate Authority (RCA). In this chapter we also introduce a Misbehaviour Authority (MA), but to minimise our notation we allow the EA to additionally assume the role of the MA.

In comparison to the ETSI reference vehicle architecture (e.g see Figure 5.1) used in the previous chapter, we adopt the standard DAA member notation such that the vehicle OBU is termed the host and the vehicle TE is termed the TPM. As before, the role of the EA is the long-term authentication of vehicles and the role of the AA is to authorise vehicles to use a particular application or service. We make the realistic assumption that vehicle hosts have an approximately synchronised clock source (e.g. from GPS) and we assume a global pseudonym change policy that divides the future into a number of epoch periods ep and defines a global pseudonym overlap period T_{overlap} during which the pseudonym of both the current and next epoch period is valid.

6.6 A Stronger Threat Model

In this work we develop a V2X scheme that is secure under a stronger threat model than the ETSI standard [85]. In particular, for unlinkability (privacy) we only require that the vehicle host is honest. We do not require that the TPM is trusted for privacy as, for our DAA instantiation, we adopt the modified TPM interface of Camenisch et al. [38] which prevents a subverted TPM from compromising the privacy of the host. In addition we allow that the EAs and AAs may all be subverted and may collaborate. This is the strongest vehicle threat model for privacy, since a compromised vehicle host can always send arbitrary privacy-compromising information to an adversary.

For our formalisation of authentication (security) we require that the TPM is uncompromised but allow for a corrupted vehicle host. The EA and the AA must be trusted for authentication as they can register any compromised vehicle they desire, but we require

that they cannot forge messages from any uncompromised TPM.

6.7 Preliminaries

This section provides the preliminary definitions that are required to fully describe our VDAA scheme. In particular we introduce our alternative ECDSA notation and then formally define the DAA signature scheme and the TPM interface that are used by our protocols.

6.7.1 ECDSA signature scheme

In both the leading European and U.S. standards, and across much of the literature, ECDSA signatures are used to provide authentication of the CAM broadcast by vehicles. The ECDSA signature scheme is fully defined in Section 5.6 of the previous chapter. Here we introduce an alternative notation that allows the reader to more readily distinguish between the use of ECDSA and DAA in our protocols. First let (d, P) denote an ECDSA public key pair output by the key generation algorithm G and let $\tau = (r, s)$ denote an ECDSA signature tuple output by the signing algorithm S when given a signing key d and message m as input. Then, where ECDSA is defined as a triple of algorithms (G, S, V) as defined in Section 5.6, we let:

1. $\mathsf{DSAGen}(1^{|n|}) = G(1^{|n|})$
2. $\mathsf{DSASign}(d, m) = S(d, m)$
3. $\mathsf{DSAVerify}(P, m, \tau) = V(P, m, \tau)$

6.7.2 DAA Intuition

Here we provide an intuition for DAA and outline how, and why, we apply it to V2X in our VDAA scheme. DAA is a cryptographic protocol for platforms that comprise a host and a TPM chip. Most commonly, DAA is used to allow a TPM chip to "attest" to the state of its host to a remote verifier. These attestations are essentially anonymous group

signatures on the current host state which are used to convince a remote verifier that the host is running signed software upon trusted hardware. Crucially, DAA is designed to allow a platform to produce these attestations in a privacy-preserving way. In particular, DAA signatures are anonymous and have user-controlled unlinkability. This means that while the verifier can confirm that a DAA signature originates from an uncompromised platform, it does not learn the identity of the platform or recognise that multiple signatures originate from the same source [40]. The user of a host system can choose whether to make multiple signatures linkable to a common source by controlling the basename bsn value that each signature is associated with. It is said [31] that a DAA signature is under the private keys of the host and associated with a basename. Association reflects the fact that the basename is not signed using the same group signature scheme as the message. A more efficient group operation suffices for the host-controlled linkability security property that the basename provides.

Briefly, all existing DAA protocols have the following structure [40]. Initially the TPM generates a secret key for which it receives a blindly-signed membership credential from a trusted issuer. A DAA credential is used to authorise signatures from a platform, with respect to a trusted issuer, to some verifier. To sign a message, the platform host and TPM work together to create a signature Proof of Knowledge SPK^* of the TPM's secret key and the host's membership credential. The final DAA signature that is sent to a verifier is a tuple that includes a re-randomised membership credential and the proofs created by the platform. DAA is not well suited to directly authenticating the CAM broadcast by vehicles because short-range linkability of these messages is a functional requirement. In addition, verifying the signature Proof of Knowledge is not well suited to the extremely low verification latencies demanded by the V2X application and the proofs required for implementing signature-based DAA revocation are also prohibitively expensive. Indeed, the signature-based DAA revocation overhead grows linearly in the size of the revocation

list [154].

In our VDAA scheme, vehicles establish long-term DAA credentials which are used to request the short-lived ECDSA pseudonym certificates that authenticate each CAM. For this purpose DAA has exactly the anonymity, user-controlled unlinkability and revocation properties that are needed to preserve vehicle privacy despite subverted and colluding certificate authorities in V2X. In addition, the ETSI standard already calls for a secure hardware cryptoprocessor onboard each vehicle and so a DAA platform readily satisfies the required vehicle architecture. Our approach allows vehicles to request regular, standards-compliant ECDSA pseudonym certificates whilst additionally providing the unlinkability of these requests that DAA permits. User-controlled unlinkability is not yet used in the protocols we present here but allows our scheme to readily support ITS services beyond cooperative awareness, such as automatic toll-road payments, which may call for linkability. We use signature-based DAA revocation, and lists linking pseudonym values to DAA signatures, to retain centralised revocation capabilities. We also introduce a new secret attribute in each DAA credential which prevents Sybil attacks.

Beyond the standard key generation, sign and verify algorithms that comprise all digital signature schemes [111], DAA also includes a join algorithm which allows a platform to join the group maintained by a trusted issuer and a link algorithm which computes whether two signatures are associated with the same basename. In VDAA, the DAA issuer is the EA which manages each vehicles long-term registration and enrolment. Correspondingly, when running the our DAA join algorithm, vehicles join the group of all registered vehicles which is managed by the EA. For the cooperative awareness application of VDAA presented in the remainder of this chapter, linkability is not desirable and so the link algorithm will always return false. In any case, regardless of whether DAA is used the host can always choose to make vehicle signatures linkable by including identifying information in the messages they authenticate.

6.7.3 DAA Formalisation

In this section we review the DAA formalisation proposed in [31]. A DAA scheme entails a set of Issuers $\mathfrak{I}$, a set of signers $\mathfrak{S}$ and a set of verifiers $\mathfrak{V}$. Each signer $(\mathfrak{t}, \mathfrak{h}) \in \mathfrak{S}$ comprises a host platform $\mathfrak{h}$ and its TPM $\mathfrak{t}$. A DAA scheme $\mathcal{DAA}$ consists of the following five efficient algorithms and protocols:

DAASetup On input the security parameter $1^{|n|}$ the issuer $\mathfrak{i} \in \mathfrak{I}$ generates a random secret key isk, the corresponding group public key ipk and the global public parameters par.

DAAJoin This protocol is run between an issuer $\mathfrak{i} \in \mathfrak{I}$ and a signer $(\mathfrak{t}, \mathfrak{h}) \in \mathfrak{S}$. The protocol creates the secret key tsk on the TPM $\mathfrak{t}$ and the DAA credential cre on the host $\mathfrak{h}$. The DAA credential cre will be used to authorise signatures from the signer $(\mathfrak{t}, \mathfrak{h})$, with respect to the trusted issuer $\mathfrak{i}$, to some verifier $\mathfrak{v} \in \mathfrak{V}$. cre is associated with the TPM secret key tsk and optionally certifies a number of attributes $\mathsf{attr} = (a_1, \ldots, a_L)$.

DAASign This algorithm is run between a host $\mathfrak{h}$ and its TPM $\mathfrak{t}$. On input the TPM secret key tsk, the basename bsn, the message m and optionally the attributes attr or the verifier nonce n_v for freshness, the platform comprising $(\mathfrak{t}, \mathfrak{h})$ uses this algorithm to produce a DAA signature σ on m and under $(\mathsf{tsk}, \mathsf{cre}, \mathsf{attr})$ and associated with bsn. The basename bsn is used by the host to control linkability between different DAA signatures.

DAAVerify On input the message m, the basename bsn, the DAA signature σ and the signature revocation list $\mathsf{Sig\text{-}RL}$, the algorithm returns either **true** (accept) or **false** (reject).

DAALink On input two DAA signatures σ_a and σ_b associated with the basenames bsn_a and bsn_b respectively, the verifier $\mathfrak{v} \in \mathfrak{V}$ uses this algorithm to determine whether

σ_a and σ_b are linked to the same host. If either σ_a or σ_b is an invalid signature then $\mathsf{DAALink}$ returns $\perp$, otherwise the algorithm returns **linked** if $\mathsf{bsn}_a = \mathsf{bsn}_b$ and **unlinked** if $\mathsf{bsn}_a \neq \mathsf{bsn}_b$.

6.7.4 TPM Interface

The TPM is an international standard for a hardware security chip that can be used to manage cryptographic keys and for remote attestation. TPMs provide a generic interface, which we detail here for completeness, that a host interacts with when executing a DAA protocol. The TPM interface has undergone a number of revisions. The version we review here is based on the work on Camenisch et al. [38] which closely follows the TPM 2.0 specification [237] but additionally includes hash-based generators that avoid the static Diffie-Hellman oracle problem [2].

Briefly there are four relevant commands that the TPM interface provides. New keys are first created using the `TPM.Create` command. Then, to sign a message the host calls `TPM.Commit`, `TPM.Hash` and finally `TPM.Sign`. The signing command is split into `TPM.Commit` and `TPM.Sign`, which produce a commitment and finalise the signing respectively, to allow for a TPM interface that supports several different signature schemes and instantiations [51]. The signature output by the `TPM.Sign` command is not a complete DAA signature but only the TPM contribution to which the host must add.

The TPM has a fixed generator $\overline{g}$ and we denote two random oracles $H : \{0,1\}^* \to \mathbb{Z}_n$, $H_{\mathbb{G}_1} : \{0,1\}^* \to \mathbb{G}_1$. The TPM initialises the set $\mathsf{Committed} = \emptyset$, the counter $\mathsf{commitId} = 0$ and provides an interface to the following four algorithms:

`TPM.Create` Selects $\mathsf{tsk} \leftarrow \mathbb{Z}_n$, computes $\mathsf{tpk} = \overline{g}^{\mathsf{tsk}}$ and outputs the public key tpk. The private key tsk is stored.

`TPM.Commit` Takes as input the secret key tsk, the optional basenames $\mathsf{bsn}_E, \mathsf{bsn}_L$ and computes the first part of the signing operation. bsn_E, if present, is used to compute

the "cleared" generator $\tilde{g} = H_{\mathbb{G}_1}(\mathsf{bsn}_E)$ and bsn_L is used to compute the generator $j = H_{\mathbb{G}_1}(\mathsf{bsn}_L)$. The first part of the signing operation is computed as follows:

1. If $\mathsf{bsn}_E \neq \perp$, set $\tilde{g} = H_{\mathbb{G}_1}(\mathsf{bsn}_E)$. Otherwise set $\tilde{g} = \overline{g}$.
2. Select $r \leftarrow \mathbb{Z}_n$, $n_t \leftarrow \{0,1\}^\eta$ and append $(\mathsf{commitId}, r, n_t)$ to $\mathsf{Committed}$.
3. Set $\overline{n}_t = H(\text{"}nonce\text{"}, n_t), E = \tilde{g}^r$ and $K, L = \perp$.
4. If $\mathsf{bsn}_L \neq \perp$, set $j = H_{\mathbb{G}_1}(\mathsf{bsn}_L)$, $K = j^{\mathsf{tsk}}$ and $L = j^r$

The TPM outputs the commitment $(\mathsf{commitId}, \overline{n}_t, E, K, L)$ and increments $\mathsf{commitId}$.

`TPM.Hash` Takes as input the messages m_t and m_h. m_t is the message the TPM attests to and m_h is the message the host adds to the attestation. If $m_t \neq \perp$, the TPM checks whether it wants to attest to m_t. If the TPM agrees then the algorithm computes $c = H(\text{"}TPM\text{"}, m_t, m_h)$, the digest c is marked "safe to sign" and the output is c.

`TPM.Sign` Takes as input the randomness reference $\mathsf{commitId}$, the digest c, the host nonce contribution n_h and completes the signing operation as follows:

1. Retrieve and remove $(\mathsf{commitId}, r, n_t)$ from $\mathsf{Committed}$.
2. Set $c' = H(\text{"}FS\text{"}, n_t \oplus n_h, c)$ and $s = r + c' * \mathsf{tsk}$. The output is (n_t, s).

Overall, when computing a DAA signature using the TPM interface above, the host learns $\mathsf{commitId}, \overline{n}_t, E, K, L, c, n_t$ and s. The purpose of these values is as follows:

1. $\mathsf{commitId}$ is the index of the randomness r that was generated by the TPM in a particular call to `TPM.Commit`. This index is needed, and is required as input, when completing the signature by calling `TPM.Sign`. The TPM builds the set $\mathsf{Committed}$ which relates each $\mathsf{commitId}$ to the random value r which was generated by the TPM. $\mathsf{commitId}$ is incremented for every signature, and the corresponding $\mathsf{Committed}$ element removed, to ensure that the same randomness is never used twice.

2. E is used to provide user-controlled linkability based on the basename bsn_E that is input to the `TPM.Commit` algorithm. Two signatures with the same bsn_E can be linked to a common source by the verifier.
3. K is used to provide private-key based revocation in DAA. TPMs that have been compromised have their exposed private keys distributed to verifiers. The value K is used by a verifier to check that a signature does not originate from a compromised TPM.
4. L is used to ensure that the private-key commitment value K can be randomised for each signature. In addition, bsn_L allows the generator for L to be chosen by the host and provides user-controlled linkability using a different generator to E. This allows for granularity over linkability for revocation and verification.
5. c is the hash of the message to be signed.
6. n_t is the TPM nonce contribution that is required to ensure the host cannot forge the SPK^* that will be created by the $\mathsf{DAASign}$ algorithm.
7. s is the final TPM signature s-value.

6.8 The VDAA Scheme

This section presents the full details of our VDAA scheme. VDAA harmonises the strong privacy guarantees of DAA with the low-latency, small signature size and standards-compliance of ECDSA signatures. In VDAA, vehicles are fitted with a TPM and use DAA as the basis of their long-term enrolment. Uniquely in our scheme, the privacy of vehicles is preserved despite colluding certificate authorities and a subverted vehicle TPM. We maintain privacy under a very strong model in which only the vehicle host needs to be fully trusted. In addition, this is accomplished whilst retaining the centralised control over vehicle revocation that is necessary for V2X. To relax the requirements for clock synchronisation, we assume a globally defined pseudonym change policy that divides the

future into a number of epoch periods ep and a global pseudonym overlap period T_{overlap} during which the pseudonym of both the current and next epoch is valid.

The intuition for our scheme is as follows. Every vehicle comprises a TPM and a host which generate a split DAA key pair $\mathsf{vk} = (\mathsf{vpk}, \mathsf{vsk})$. Vehicles join the scheme by obtaining a partially blind DAA signature $\mathsf{cre} = \mathsf{PBSig}(\mathsf{isk}, \mathsf{vpk})$ on the split public key vpk from the EA. To obtain ECDSA pseudonym certificates, vehicles make an anonymous request for each epoch ep by using the DAA sign algorithm to authenticate to the AA. To prevent the abuse of anonymous credentials, a unique serial token ser is included in each request. Serial tokens are derived from the Sybil secret s which is unique to each vehicle and the requested pseudonym validity epoch ep. Serial tokens prevent Sybil attacks as any vehicle that makes multiple requests for pseudonyms in the same epoch is forced to do so with the same serial token and therefore can be denied additional credentials.

The AA maintains a list of DAA signature and ECDSA pseudonym tuples, Auth-L, which enables vehicles that send malicious messages to be removed by denying them new credentials in the future. Broadcast message signing and verification are just the standard ECDSA operations from Section 6.7.1, which both maintains the performance that is necessary for safety-critical V2X applications and ensures that a subverted TPM cannot compromise the privacy of the vehicle. Whilst VDAA can be instantiated using either a LRSW [166] or q-Strong Diffie Hellman (q-SDH) [27] based DAA scheme, in the remainder of this section and our analysis we focus on the q-SDH based scheme of Camenisch et al. [38]. The q-SDH DAA scheme we use has a more efficient attribute certification mechanism which we use to prevent Sybil attacks.

The VDAA scheme consists of 3 algorithms and 3 protocols. The Setup algorithm is run once by the EA and the AA to generate the scheme public and private parameters. The Join protocol is typically run once for each vehicle that joins the scheme and the Issue protocol is run each time that a vehicle requires a pseudonym certificate for a particular

epoch. Vehicles sign and verify broadcast messages using the Sign and Verify algorithms, respectively, and the Revocation protocol is run when removing misbehaving vehicles.

Setup

The VDAA setup algorithm is run once to initialise the parameters of the scheme. On input the security parameter 1^η the EA selects the group public key pair $\mathsf{ik} = (\mathsf{ipk}, \mathsf{isk})$ and the public parameters par which include the vehicle revocation list $\mathsf{Pub\text{-}RL} = \emptyset$ and the private-key revocation list $\mathsf{Priv\text{-}RL} = \emptyset$. In particular ik is a BBS+ signature scheme [9] key pair which is generated as follows:

1. Choose uniformly at random generator $h \leftarrow \mathbb{G}_1$ and the private key $x \leftarrow \mathbb{Z}_n$.
2. Set $X = g_2^x$ and $X' = g_1^x$.
3. Prove $\pi_{\text{ipk}} = \mathsf{SPK}[x : X = g_2^x \wedge X' = g_1^x](\text{“setup”})$.
4. Let $\mathsf{ipk} = (h, X, X', \pi_{\text{ipk}})$ and $\mathsf{isk} = x$.

The AA selects the ECDSA public key pair $\mathsf{ak} = (P_{\text{AA}}, d_{\text{AA}})$ and creates the signature revocation list $\mathsf{Sig\text{-}RL} = \emptyset$, the attestation list $\mathsf{Auth\text{-}L} = \emptyset$ and the serial token list $\mathsf{Ser\text{-}L} = \emptyset$. Specifically ak is an ECDSA key pair that is output by the DSAGen algorithm defined in Section 6.7.1.

Join

The first step of VDAA is the Join protocol, shown in Figure 6.1, during which a vehicle joins the EA membership group. Our Join protocol is primarily based on the DAA Join protocol of Camenisch et al. [38], which we adapt to include our Sybil attack resistance mechanism and revocation capabilities. For simplicity we assume that the vehicle host manufacturer is also the EA and can therefore be certain that it is executing the protocol with a genuine TPM. The Join protocol can also be run after the vehicle host has been shipped, for which we assume that the vehicle host has a certified endorsement key installed

and that the corresponding certificate is available to the EA. The EA takes as input the group public key pair $\mathsf{ik} = (\mathsf{ipk}, \mathsf{isk})$ and the public and private revocation lists $\mathsf{Pub\text{-}RL}$ and $\mathsf{Priv\text{-}RL}$, respectively. The vehicle host takes as input ipk and then the Join protocol is as follows:

1. The vehicle host requests to join the VDAA scheme and the EA responds with the nonce $n \leftarrow \{0,1\}^{\eta}$ for freshness.
2. The vehicle host requests the TPM to create a new DAA key pair. The TPM selects the DAA key pair $\mathsf{tk} = (\mathsf{tpk}, \mathsf{tsk})$, stores the private key $d_{\mathfrak{t}}$ and sends the public key tpk to the vehicle host.
3. The vehicle host forwards the nonce n to the TPM and then requests the split vehicle key contribution $\mathsf{tpk} = \overline{g}^{\mathsf{tsk}}$ and the proof $\pi_{\mathsf{tpk}} = \mathsf{SPK}^{*}[\mathsf{tsk} : \mathsf{tpk} = \overline{g}^{\mathsf{tsk}}](\text{“join”}, n)$ which asserts that:
   i. The TPM has the private key tsk corresponding to tpk.
   ii. The TPM generated the split vehicle key contribution $\mathsf{tpk}' = \overline{g}^{\mathsf{tsk}}$ such that it corresponds to tsk.

   The TPM computes tpk', the proof π_{tpk} and sends them to the vehicle host.
4. The vehicle host selects the split vehicle key contribution $\mathsf{hsk} \leftarrow \mathbb{Z}_n$, computes the vehicle public key $\mathsf{vpk} = \mathsf{tpk}' \cdot \overline{g}^{\mathsf{hsk}}$ and the proof $\pi_{\mathsf{vpk}} = \mathsf{SPK}[\mathsf{hsk} : \mathsf{vpk}/\mathsf{tpk}' = \overline{g}^{\mathsf{hsk}}](\text{“}join\text{”}, n)$ which asserts that vpk is a signature proof of knowledge SPK on n. The vehicle host also selects the Sybil secret s and computes the public key $\mathsf{spk} = \overline{g}^{s}$ which is included in the request for group membership. The vehicle host sends $\mathsf{tpk}, \mathsf{tpk}', \mathsf{vpk}, \mathsf{spk}, \pi_{\mathsf{tpk}}$ and π_{vpk} to the EA.
5. The EA verifies that vpk is not in the vehicle revocation list $\mathsf{Pub\text{-}RL}$ and that it does not correspond to any revoked private key in $\mathsf{Priv\text{-}RL}$. Next, the EA verifies the proofs $\pi_{\mathsf{tpk}}, \pi_{\mathsf{vpk}}$ and then computes the membership credential cre using a partially blind signature PBSign that certifies vsk by signing vpk. The resulting DAA

credential $\mathsf{cre} = \mathsf{PBSig}(\mathsf{isk}, (\mathsf{vpk}, \mathsf{spk}))$ is sent to the vehicle host. Specifically, cre is a blindly-signed BBS+ signature on the message (vsk, s) which is computed as follows:

i. Choose $(e, r) \leftarrow \mathbb{Z}_n^2$.

ii. Compute $A = (g_1 \cdot h^r \cdot \mathsf{vpk} \cdot \mathsf{spk})^{\frac{1}{e+x}}$.

iii. Set $\mathsf{cre} = (A, e, r)$.

6. The vehicle host verifies the DAA credential cre with respect to vpk, the Sybil public attribute spk and the group public key ipk. In particular the vehicle host computes $b = g_1 \cdot h^r \cdot \mathsf{vpk} \cdot \mathsf{spk}$ and checks that $e(A, X \cdot g_2^e) = e(b, g_2)$. The host stores its secret key hsk, the DAA credential cre and the Sybil secret s.

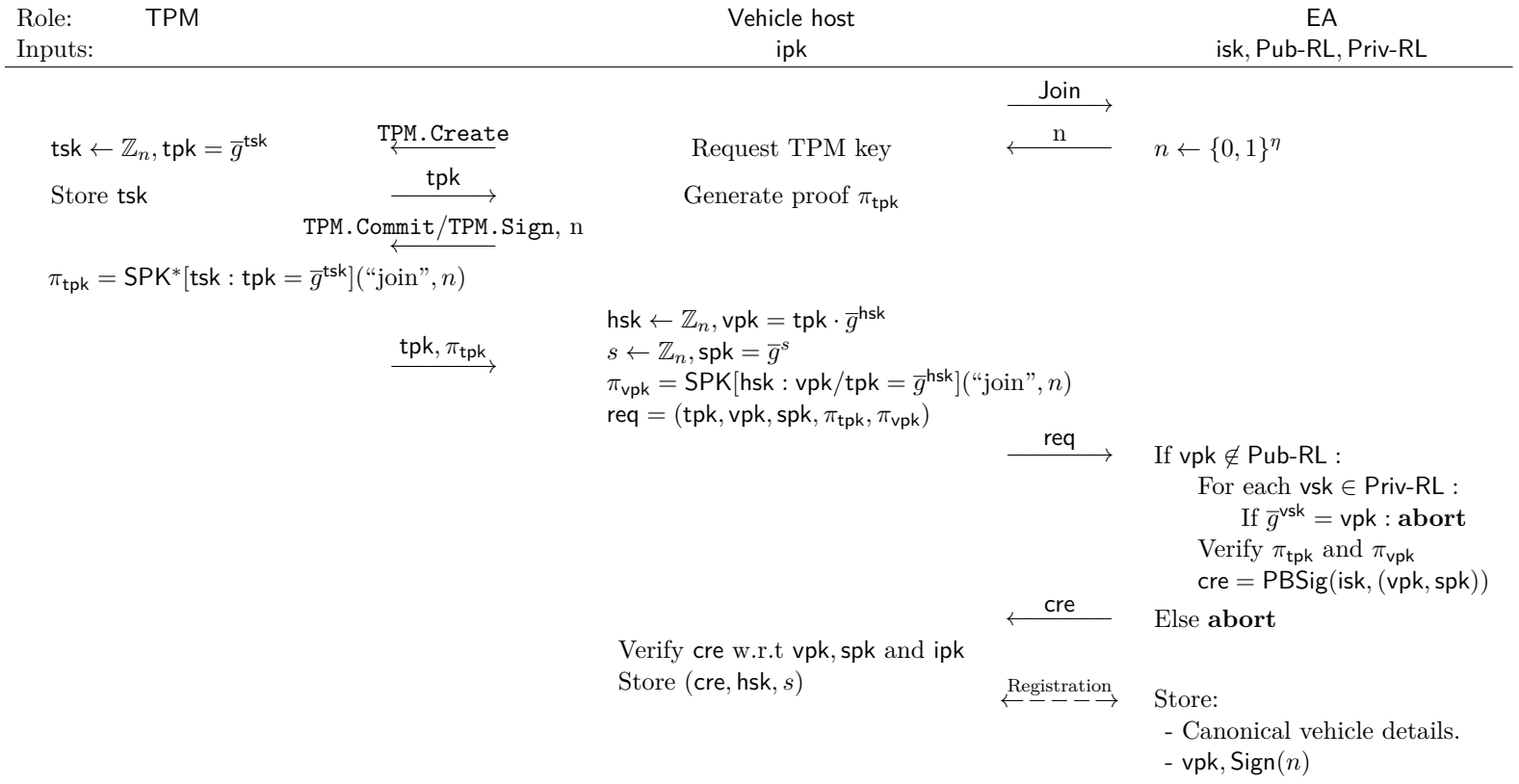


Figure 6.1: The VDAA Join protocol.

Role:	TPM	Vehicle host		AA
Inputs:	tsk	$(\mathsf{cre}, \mathsf{hsk}, s), \mathsf{ipk}, \mathsf{ep}, \mathsf{Sig\text{-}RL}$		$x_{\mathrm{AA}}, \mathsf{ipk}, \mathsf{Sig\text{-}RL}, \mathsf{Auth\text{-}L}, \mathsf{Ser\text{-}L}$
DAASign:		$\mathsf{bsn} = \perp$ $x_{\mathsf{ep}} \leftarrow \mathbb{Z}_n^*, P_{\mathsf{ep}} = x_{\mathsf{ep}} \times G$ $\mathsf{ser}_{s,\mathsf{ep}} = H_{\mathbb{G}_1}(1 \parallel \mathsf{ep})^s$ $\mathsf{rev} = H_{\mathbb{G}_1}(1 \parallel \mathsf{bsn})^{\mathsf{vsk}}$ $\pi_{\mathsf{cre}} = \mathsf{NIZK}^*[(\mathsf{vsk}, s, \mathsf{cre}) : \mathsf{ser}_{s,\mathsf{ep}} = H_{\mathbb{G}_1}(1 \parallel \mathsf{ep})^s$ $\wedge\ \mathsf{rev} = H_{\mathbb{G}_1}(1 \parallel \mathsf{bsn})^{\mathsf{vsk}}$ $\wedge\ 1 = \mathsf{PBVf}(\mathsf{ipk}, \mathsf{cre}', \mathsf{vsk}, s)](\text{“sign”}, (P_{\mathsf{ep}}, \mathsf{ep}), \mathsf{Sig\text{-}RL}))$ For each $(\mathsf{bsn}_i, \mathsf{rev}_i) \in \mathsf{Sig\text{-}RL}$: $\pi_{\mathsf{Sig\text{-}RL},i} = \mathsf{SPK}^*[\mathsf{vsk} : H_{\mathbb{G}_1}(1 \parallel \mathsf{bsn}_i)^{\mathsf{vsk}} \neq \mathsf{rev}_i \wedge \mathsf{rev} = H_{\mathbb{G}_1}(1 \parallel \mathsf{bsn}_i)^{\mathsf{vsk}}](\text{“sign”})$ $\sigma = (\mathsf{ser}_{s,\mathsf{ep}}, (\mathsf{bsn}, \mathsf{rev}), \pi_{\mathsf{cre}}, \{\pi_{\mathsf{Sig\text{-}RL},i}\})$	$\xrightarrow{(\sigma, P_{\mathsf{ep}} \parallel \mathsf{ep})}$	
DAAVerify:				Parse $\sigma = (\mathsf{ser}_{s,\mathsf{ep}}, (\mathsf{bsn}, \mathsf{rev}), \pi_{\mathsf{cre}}, \{\pi_{\mathsf{Sig\text{-}RL},i}\})$ Verify $\pi_{\mathsf{cre}}, \pi_{\mathsf{Sig\text{-}RL},i}$ w.r.t $\mathsf{ipk}, m, \mathsf{Sig\text{-}RL}$ For each $\mathsf{ser} \in \mathsf{Ser\text{-}L}$: If $\mathsf{ser}_{s,\mathsf{ep}} = \mathsf{ser}_{s,\mathsf{ep}}$: **abort** Add $\mathsf{ser}_{s,\mathsf{ep}}$ to Ser-L, Add σ to Auth-L For each $\mathsf{vsk} \in \mathsf{Priv\text{-}RL}$: If $H_{\mathbb{G}_1}(1 \parallel \mathsf{bsn})^{\mathsf{vsk}} = \mathsf{rev}$: **abort**
DSASign :			$\xleftarrow{\tau}$	$\tau = \mathsf{DSASign}(x_{\mathrm{AA}}, P_{\mathsf{ep}} \parallel \mathsf{ep})$
DSAVerify :		If $\mathsf{DSAVerify}(P_{\mathsf{ep}}, (P_{\mathsf{ep}} \parallel \mathsf{ep}), \tau) \neq \mathbf{true}$: **abort** Store $(\mathsf{ep}, x_{\mathsf{ep}}, \tau)$		

Figure 6.2: The VDAA Issue protocol.

Issue

The VDAA Issue protocol, shown in Figure 6.2, is run each time that a vehicle requires a signed pseudonym certificate for a particular epoch ep. At the beginning, the TPM has the private key tsk and the vehicle host has the secret key hsk, the DAA credential cre, the Sybil secret s, the AA group public key ipk, the epoch ep and the signature revocation list Sig-RL. The AA has the ECDSA private key d_{AA}, ipk, Sig-RL, the attestation list Auth-L and the serial token list Ser-L. Vehicles are required to periodically download the signature revocation list Sig-RL, for example nightly, and the AA will not accept any proof of non-revocation $\pi_{\mathsf{Sig\text{-}RL}}$ that does not include all of the required elements. The Issue protocol is as follows:

1. The vehicle host selects a random epoch key d_{ep} and then computes the pseudonym public key $P_{\mathsf{ep}} = d_{\mathsf{ep}} \times G$ and the serial token $\mathsf{ser}_{s,\mathsf{ep}} = H_{\mathbb{G}_1}(1 \parallel \mathsf{ep})^s$.
2. The vehicle host and TPM jointly compute the signature revocation token $\mathsf{rev} = H_{\mathbb{G}_1}(1 \parallel \mathsf{bsn})^{\mathsf{vsk}}$, the proof of membership credential π_{cre} and, for each tuple $(\mathsf{bsn}_i, \mathsf{rev}_i)$ in Sig-RL, the non-revocation proof $\pi_{\mathsf{Sig\text{-}RL},i}$. The proof π_{cre} is computed using the Prove protocol of Camenisch et al. [38] which is unforgeable, device-bound and zero-knowledge even if the TPM is subverted. In particular an SPK^* created with the Prove protocol is zero-knowledge even when the TPM is corrupt. Essentially this is because all parts of the proof are uniform in $\mathbb{Z}_n$ regardless of the randomness chosen by the TPM, since they necessarily also include randomness from the host. The DAA credential and proof of membership credential π_{cre} are computed as follows:
    i. The vehicle host re-randomises the BBS+ credential $\mathsf{cre}' = ((A, e, r), b)$ established in the Join protocol. The vehicle host chooses $q_1 \leftarrow \mathbb{Z}_n^\star, q_2 \leftarrow \mathbb{Z}_n, q_3 \leftarrow \frac{1}{q_1}$, sets $A' = A^{q_1}, \overline{A} = A'^{-e} \cdot b^{q_1}, b' = b^{q_1} \cdot h^{-q_2}$ and $r' = r - q_2 \cdot q_3$. The re-randomised credential is $\mathsf{cre}' = (\overline{A}, A', b')$.

ii. The vehicle host and TPM jointly compute the proof of membership credential π_{cre}:

$$\begin{aligned}\pi_{\mathsf{cre}} = &\mathsf{SPK}^{\star}\{(\mathsf{vsk}, s, e, q_2, q_3, r') : \\ &g_1^{-1} \cdot \overline{g}^{-s} = b'^{-q_3} \cdot h^{r'} \cdot \mathsf{vpk} \cdot \mathsf{spk} \ \wedge \\ &\mathsf{ser}_{s,\mathsf{ep}} = H_{\mathbb{G}_1}(1 \parallel \mathsf{ep})^{s} \\ &\mathsf{rev} = H_{\mathbb{G}_1}(1 \parallel \mathsf{bsn})^{\mathsf{vsk}} \ \wedge \\ &\overline{A} \cdot b'^{-1} = A'^{-e}\}(\text{“sign”}, \mathsf{Sig\text{-}RL}))\end{aligned}$$

The final DAA signature is $\sigma = (\mathsf{ser}_{s,\mathsf{ep}}, (\mathsf{bsn}, \mathsf{rev}), \mathsf{cre}', \pi_{\mathsf{cre}}, \{\pi_{\mathsf{Sig\text{-}RL},i}\})$. The vehicle host sends $(\sigma, P_{\mathsf{ep}} \parallel \mathsf{ep})$ to the AA.

3. The AA parses the DAA signature σ and verifies the proofs π_{cre} and $\{\pi_{\mathsf{Sig\text{-}RL},i}\}$ with respect to the group public key ipk, the message m and the revocation list $\mathsf{Sig\text{-}RL}$. In particular, the AA checks that $A' \neq 1$ and $e(A', X) = e(\overline{A}, g_2)$ with respect to the randomised DAA credential $\mathsf{cre}' = (\overline{A}, A', b')$ and $\mathsf{ipk} = (h, X, X', \pi_{\mathrm{ipk}})$. The AA also ensures that the serial token $\mathsf{ser}_{s,\mathsf{ep}}$ is novel and that vsk has not been revoked. The DAA signature and ECDSA pseudonym tuple $(\sigma, P_{\mathsf{ep}} \parallel \mathsf{ep})$ is added to the attestation list $\mathsf{Auth\text{-}L}$, the serial token $\mathsf{ser}_{s,\mathsf{ep}}$ is added to $\mathsf{Ser\text{-}L}$ and the host is sent the ECDSA signature τ on the requested pseudonym public key and epoch $(P_{\mathsf{ep}} \parallel \mathsf{ep})$.
4. The vehicle host verifies the AA signature τ on $(P_{\mathsf{ep}} \parallel \mathsf{ep})$ and then creates a record that links the epoch ep with the signature τ and the signing key d_{ep}.

Sign

The VDAA Sign algorithm is run each time that a vehicle host signs a broadcast message. The Sign algorithm is simply the standard ECDSA $\mathsf{DSASign}$ algorithm from Section 6.7.1 and is run by the vehicle host only. Since the TPM does not take part in the signing of

broadcast messages, it is unable to compromise the privacy of the vehicle host.

To sign a message m, the vehicle host has the ECDSA signing key x_{ep} for the current epoch ep in which the signature should be valid. The vehicle host runs the DSASign algorithm from Section 6.7.1 which computes the following:

1. Choose an instance key $k \leftarrow \mathbb{Z}_n^\star$.
2. Compute the instance curve point $R = (r_x, r_y) = k \times G$ and the integer $r = r_x \mod n$.
3. Compute $s = k^{-1} \cdot (H_q(m) + d_{\mathsf{ep}} \cdot r) \mod n$.
4. The signature on m is $\tau = (r, s)$.

The signing could be split between the vehicle host and the TPM using the IFAL public-key derivation technique from Section 5.7.2 in the previous chapter, however the TPM would be able to compromise the privacy of the vehicle host by choosing bad instance keys. An alternative technique that would allow the signing to be split between the vehicle host and the untrusted TPM is the efficient two-party ECDSA signing protocol of Lindell [162]. This would have the advantage of requiring the involvement of the TPM for creating broadcast message signatures. However, in VDAA since the TPM is necessary to request a signed pseudonym certificate, there is little to be gained unless the epoch periods are very long. In addition, two-party ECDSA is computationally far more expensive than the single party case.

Verify

The Verify algorithm, which is used to verify every broadcast message that is received by a vehicle, is unchanged from the standard ECDSA verification algorithm in Section 6.7.1. The vehicle simply runs the DSAVerify algorithm which takes as input the pseudonym public key P_{ep}, the signed message tuple (m, τ) and outputs either **true** (accept) or **false** (reject). For every unique pseudonym public key P_{ep} that is used to authorise a received

message, the vehicle host additionally verifies that there is a signature τ_{ep} on P_{ep} which is valid with respect to the AA public key P_{AA}.

Revocation

In VDAA there are three different mechanisms for revocation and correspondingly, three different protocols. VDAA supports identity-based, message-based and private-key based revocation with the following three protocols:

Identity-based revocation The first revocation protocol is based on the EA being provided with the canonical registration information of the vehicle that should be removed from the scheme. This could occur when the vehicle is taken off the road by its owner, or after the vehicle is "written off" by the insurer following an accident. The EA has the vehicle identity and then looks up the vehicle public key vpk and signature σ that were provided during the Issue protocol. The EA adds vpk to the vehicle revocation list Pub-RL and sends σ is sent to the AA. The AA adds σ to the signature revocation list Sig-RL.

Message-based revocation It is critical for V2X that dishonest vehicles which send false information can be removed from participation. In message-based revocation the AA is provided with a signature τ on a message m, the attestation list Auth-L and Sig-RL. The AA uses Auth-L to identify the DAA signature σ that was used to request the pseudonym P_{ep} with respect to τ in the Issue protocol. The AA adds σ to Sig-RL. When requesting new pseudonyms, vehicles prove in zero knowledge that they did not create any of the signatures in Sig-RL. Vehicles that have been revoked will be denied future pseudonym signatures after at most one epoch.

Private-key revocation The final revocation mechanism is based on a compromised vehicle private key that has been discovered and provided to both the EA and the AA. In the Join protocol that is run between the EA and the vehicle, the EA checks

that vpk does not correspond to any revoked private key vsk. If vsk is revoked, then vpk is added to Pub-RL. In the Issue protocol that is run between the AA and the vehicle, the revocation tuple $(\mathsf{bsn}, \mathsf{rev} = \mathbb{H}_{\mathbb{G}_1}(1 \parallel \mathsf{bsn})^{\mathsf{vsk}})$ is used to check that the vehicle is not using a revoked vsk. Vehicles with revoked private keys are denied at both the Join and Issue stages of the VDAA scheme.

6.8.1 Sybil Attack Resistance

Sybil attacks [73], in which a small number of entities impersonate multiple identities so as to compromise a disproportionate share of the system, are resisted in VDAA using the following approach. When a vehicle joins an instance of our scheme by running the Join protocol with the EA, the host selects a Sybil secret s and computes the corresponding Sybil public key $\mathsf{spk} = \overline{g}^s$. The Sybil public key spk is used by the EA to compute a blind signature on the Sybil secret s which is included in the DAA credential cre that is issued to the vehicle. In particular cre is a BBS+ signature (A, e, r) which is computed by the EA such that $A = (g_1 \cdot h^r \cdot \mathsf{vpk} \cdot \mathsf{spk})^{\frac{1}{e+x}}$ as described in the Join protocol description above.

To request a signed pseudonym certificate for a particular epoch ep, the vehicle must run the Issue protocol shown in Figure 6.2. The vehicle host first composes the serial token $\mathsf{ser}_{s,\mathsf{ep}} = H_{\mathbb{G}_1}(1 \parallel \mathsf{ep})^s$ using the epoch value ep which is chosen from a global public function that relates every epoch of time to a unique value. Indeed the entire generator of this expression $H_{\mathbb{G}_1}(1 \parallel \mathsf{ep})$ is a global public value for each epoch value ep in the image of the epoch function.

Where cre' is the re-randomised BBS+ credential calculated using the method described in the Issue protocol, the vehicle host and TPM compute together the proof of membership credential using the following general technique:

$$\pi_{\mathsf{cre}} = \mathsf{NIZK}^*[(\mathsf{vsk}, s, \mathsf{cre}) : \mathsf{ser}_{s,\mathsf{ep}} = H_{\mathbb{G}_1}(1 \parallel \mathsf{ep})^s \\ \wedge\ \mathsf{rev} = H_{\mathbb{G}_1}(1 \parallel \mathsf{bsn})^{\mathsf{vsk}} \\ \wedge\ 1 = \mathsf{PBVf}(\mathsf{ipk}, \mathsf{cre}', \mathsf{vsk}, s)](\text{“sign”}, (P_{\mathsf{ep}}, \mathsf{ep}), \mathsf{Sig\text{-}RL}))$$

In other words, the vehicle platform proves in zero knowledge that the serial token $\mathsf{ser}_{s,\mathsf{ep}}$ is constructed properly using the Sybil secret s, and that s is also the same value which was blindly signed by the EA when running the Join protocol. Since the AA also keeps track of all of the serial tokens that is has seen, any vehicle that attempts to request two pseudonyms which are simultaneously valid will forfeit its unlinkability.

6.9 VDAA Formalisation

This section presents our formal definition of a VDAA scheme and the associated unforgeability and unlinkability requirements for V2X.

6.9.1 VDAA Scheme

In VDAA, each vehicle $V_i = (\mathfrak{h}_i, \mathfrak{t}_i)$ comprises a vehicle host $\mathfrak{h}_i$ and its TPM $\mathfrak{t}_i$. With reference to the definition of a DAA scheme in Section 6.7.3, in VDAA the EA is the issuer, the AA is the verifier and each vehicle is a signer. The AA maintains the list of vehicle DAA signatures Auth-L and the list of serial tokens Ser-L. In addition, the EA manages three revocation lists: the vehicle revocation list Pub-RL, the signature revocation list Sig-RL and the private-key revocation list Priv-RL. A VDAA scheme comprises three efficient algorithms Setup, Verify, Revoke and three interactive protocols Join, Issue and Sign.

Setup The EA takes as input the security parameter 1^η and outputs the group public key pair $\mathsf{ik} = (\mathsf{ipk}, \mathsf{isk})$ and the global public parameters par which include revocation lists $\mathsf{Pub\text{-}RL} = \emptyset$ and $\mathsf{Priv\text{-}RL} = \emptyset$. The AA and outputs the ECDSA public key pair $\mathsf{ak} = (P_{\mathrm{AA}}, d_{\mathrm{AA}})$, the signature revocation list $\mathsf{Sig\text{-}RL} = \emptyset$, the attestation list

$\mathsf{Auth\text{-}L} = \emptyset$ and the serial token list $\mathsf{Ser\text{-}L} = \emptyset$.

Join is run between the EA and a vehicle $V_i = (\mathfrak{h}_i, \mathfrak{t}_i)$. The EA is given the group public key pair $\mathsf{ik} = (\mathsf{ipk}, \mathsf{isk})$ and V_i is given ipk. Eventually, $\mathfrak{t}_i$ outputs a private key tsk. The host $\mathfrak{h}_i$ will output a secret key hsk, DAA credential cre and Sybil secret s. A revoked vehicle will output nothing $\perp$.

Issue is run between the AA and the vehicle $V_i = (\mathfrak{h}_i, \mathfrak{t}_i)$. The AA is given the group public key ipk, the ECDSA private key d_{AA}, the attestation list $\mathsf{Auth\text{-}L}$ and the serial token list $\mathsf{Ser\text{-}L}$. The TPM $\mathfrak{t}_i$ has the private key tsk and the host $\mathfrak{h}_i$ has ipk, the private key hsk, the DAA credential cre, the Sybil secret s and the epoch ep. Eventually, the AA will output the updated $\mathsf{Auth\text{-}L'}$, the updated $\mathsf{Ser\text{-}L'}$ and either the pseudonym signature τ (accept) or $\perp$ (reject).

Sign is run by a vehicle V_i and takes as input the ECDSA public-key pair $(d_{\mathfrak{h}}, P_{\mathfrak{h}})$, the message m and the epoch ep. Eventually, V_i outputs the ECDSA signature $\tau = (r, s)$ on m with respect to the pseudonym public key $P_{\mathsf{ep}} = d_{\mathsf{ep}} \times G$.

Verify is run by a vehicle V_i, takes as input the ECDSA signed message (m, τ) and outputs either **true** (accept) or **false** (reject).

Revoke has three different implementations. For vehicle based revocation, the EA takes as input the vehicle public key vpk and outputs the updated vehicle revocation list $\mathsf{Pub\text{-}RL'}$. Signature based revocation is run between the AA and the EA. The AA takes as input the group public key ipk, the signed message (m, τ), the signature revocation list $\mathsf{Sig\text{-}RL}$ and the attestation list $\mathsf{Auth\text{-}L}$. The AA sends the corresponding DAA signature (m', σ) to the EA which outputs the updated $\mathsf{Sig\text{-}RL'}$. For private-key based revocation the EA takes as input ipk, the private-key list $\mathsf{Priv\text{-}RL}$, the vehicle private key vsk and outputs the updated list $\mathsf{Priv\text{-}RL'}$.

6.9.2 VDAA Security and Privacy

In this section we formalise the security and privacy of a VDAA scheme based on the standard V2X requirements from Section 6.4. In the previous chapter we formalised these requirements in terms of a secure and a "privacy conscious" V2X scheme. In this chapter, we develop two intuitively-similar game based security definitions but that additionally take account of the extra DAA parameters and security guarantees introduced for VDAA.

Unforgeability

Intuitively, we require that a vehicle can determine the authenticity of each broadcast message that is received. If all TPMs, the EA and the AA are uncorrupted then no adversary should be able to either create a valid request for a pseudonym certificate or create a valid ECDSA signature on any message. We capture this requirement by defining the unforgeability game **Forge-Game** which takes as input the security parameter 1^η and an efficient adversary $\mathcal{A}$ who interacts with a challenger $\mathcal{C}$. The unforgeability game is as follows:

Forge-Game$_{\mathcal{C}}(1^\eta, \mathcal{A})$:

---

1. The challenger $\mathcal{C}$ simulates the $\mathsf{Setup}(1^\eta)$ algorithm which outputs the EA public key pair $\mathsf{ik} = (\mathsf{ipk}, \mathsf{isk})$, the parameters par and the AA ECDSA public key pair $\mathsf{ak} = (P_{\mathrm{AA}}, d_{\mathrm{AA}})$. $\mathcal{C}$ simulates N_V vehicles with identities $\{V_1, \ldots, V_{N_V}\}$ and also simulates the EA and the AA including the Join and Issue protocols from Section 6.9.1. Finally, $\mathcal{C}$ provides the adversary $\mathcal{A}$ with the public parameters $(\mathsf{ipk}, \mathsf{par}, P_{\mathrm{AA}})$ and a reference to each vehicle $V_i \in \{V_1, \ldots, V_{N_V}\}$.

2. The challenger $\mathcal{C}$ simulates each vehicle $V_i \in \{V_1, \ldots, V_{N_V}\}$ by selecting the vehicle secret key vsk_i, the Sybil secret s_i and by simulating the Join protocol and the EA so that each vehicle V_i has the DAA credential cre_i on vsk_i and s_i.

3. Challenge: Polynomially many times, adversary $\mathcal{A}$ requests challenger $\mathcal{C}$ to sign a message m in epoch ep on behalf of vehicle V_i. $\mathcal{C}$ simulates the vehicle V_i and then:

i. If V_i does not have the epoch key d_{ep} then $\mathcal{C}$ simulates the Issue protocol by selecting the random epoch key d_{ep} and then computing the ECDSA signature τ_i on the pseudonym public key $P_{\mathsf{ep},i}$ with respect to the AA public key P_{AA}.

ii. $\mathcal{C}$ simulates the $\mathsf{DSASign}$ algorithm and provides $\mathcal{A}$ with the ECDSA signature τ on message m with respect to pseudonym $P_{\mathsf{ep},i}$.

4. Output: The adversary $\mathcal{A}$ outputs the signature $\tau^\star$, the message $m^\star$, the pseudonym public key $P^\star_{\mathsf{ep}}$ and the AA signature ${\tau_{\mathsf{ep}}}^\star$.

An adversary $\mathcal{A}$ wins the unforgeability game if and only if the following three conditions hold true:

1. The message $m^\star$ does not correspond to any query m made by the adversary $\mathcal{A}$ to the challenger $\mathcal{C}$.
2. The tuple $(m^\star, \tau^\star)$ is a valid message-signature pair with respect to the pseudonym public key ${P_{\mathsf{ep}}}^\star$. In other words $\mathsf{DSAVerify}({P_{\mathsf{ep}}}^\star, m^\star, \tau^\star) = \mathbf{true}$.
3. The pseudonym public key and AA signature $({P_{\mathsf{ep}}}^\star, {\tau^\star}_{\mathsf{ep}})$ is a valid message-signature pair with respect to the AA public key P_{AA}. i.e. $\mathsf{DSAVerify}(P_{\mathrm{AA}}, {P_{\mathsf{ep}}}^\star, {\tau^\star}_{\mathsf{ep}}) = \mathbf{true}$.

Definition 6.9.1 Let $\mathcal{A}$ denote an adversary that plays the **Forge-Game**. We denote by $\mathbf{Adv}[\mathcal{A}^{\mathrm{forge}}_{\mathsf{VDAA}}] = \Pr[\mathcal{A} \text{ wins}]$ the advantage with which the adversary $\mathcal{A}$ breaks the unforgeability game. We say that a VDAA scheme is unforgeable if for all efficient adversaries $\mathcal{A}$, $\mathbf{Adv}[\mathcal{A}^{\mathrm{forge}}_{\mathsf{VDAA}}]$ is negligible.

Unlinkability

Informally, one vehicle that signs V2X messages using two different ECDSA pseudonyms during two non-overlapping epochs should be indistinguishable from if two distinct vehicles had signed the same messages. As we cannot cryptographically defend from message contents which reveal linkable information, we focus on ensuring that the signatures themselves do not undermine the privacy of a vehicle. We capture this requirement by defining

the unlinkability game **Priv-Game** which takes as input the security parameter 1^η and an efficient adversary $\mathcal{A}$ who interacts with a challenger $\mathcal{C}$. The unlinkability game is as follows:

Priv-Game$_{\mathcal{C}}(1^\eta, \mathcal{A})$:

---

1. The challenger $\mathcal{C}$ simulates the $\mathsf{Setup}(1^\eta)$ algorithm and provides the adversary $\mathcal{A}$ with the resulting $\mathsf{ik} = (\mathsf{ipk}, \mathsf{isk})$, $\mathsf{ak} = (P_{\mathrm{AA}}, d_{\mathrm{AA}})$ and the parameters par. $\mathcal{C}$ also simulates 2 vehicles with identities V_0 and V_1.

2. Let the number of vehicles $N_V = 2$, then this step is the same as in the unforgeability game **Forge-Game**.

3. The adversary $\mathcal{A}$ selects two distinct epochs $\mathsf{ep0}, \mathsf{ep1}$ and submits them to $\mathcal{C}$.

4. Challenger $\mathcal{C}$ flips a bit $b \leftarrow \{0, 1\}$. For each vehicle $V_i \in (V_b, V_{b-1})$, $\mathcal{C}$ simulates V_i and selects two distinct epoch keys $d_{\mathsf{ep0},i}$ and $d_{\mathsf{ep1},i}$. For each corresponding pseudonym public key $P_{\mathsf{ep0},i}$ and $P_{\mathsf{ep1},i}$, $\mathcal{C}$ simulates the Issue protocol with the adversary $\mathcal{A}$ who simulates the AA. $\mathcal{C}$ acquires the ECDSA signatures $\tau_{\mathsf{ep0},i}, \tau_{\mathsf{ep1},i}$ with respect to the AA public key P_{AA} on $P_{\mathsf{ep0},i}$ and $P_{\mathsf{ep1},i}$.

5. Challenge: Polynomially many times, the adversary $\mathcal{A}$ requests the challenger $\mathcal{C}$ to sign a message m during epoch ep.

   - If $\mathsf{ep} \notin \{\mathsf{ep0}, \mathsf{ep1}\}$ the challenger $\mathcal{C}$ outputs $\perp$.
   - If $b = 0$, $\mathcal{C}$ simulates vehicle V_0, simulates the $\mathsf{DSASign}$ algorithm and outputs the signed message (τ, m) with respect to pseudonym $P_{\mathsf{ep0},0}$.
   - If $b = 1$ and $\mathsf{ep} = \mathsf{ep0}$ then $\mathcal{C}$ simulates vehicle V_0. If $\mathsf{ep} = \mathsf{ep1}$ then $\mathcal{C}$ simulates V_1. $\mathcal{C}$ outputs the signed message (τ, m) with respect to pseudonym public key $P_{\mathsf{ep0},0}$ or $P_{\mathsf{ep1},1}$, respectively.

6. Output: The adversary $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$ indicating its guess of b.

An adversary $\mathcal{A}$ wins the unlinkability game if $b = b'$.

Definition 6.9.2 Let $\mathcal{A}$ denote an adversary that plays the **Priv-Game**. We denote by $\mathbf{Adv}[\mathcal{A}^{\text{link}}_{\mathsf{VDAA}}] = |\Pr[b' = b] - \frac{1}{2}|$ the advantage with which the adversary $\mathcal{A}$ breaks the unlinkability game. We say that a VDAA scheme is unlinkable if for all efficient adversaries $\mathcal{A}$, the advantage $\mathbf{Adv}[\mathcal{A}^{\text{link}}_{\mathsf{VDAA}}]$ is negligible.

6.10 The Security and Privacy of VDAA

This section shows that our VDAA scheme that we present in Section 6.8 is secure with respect to unforgeability and unlinkability as defined in Section 6.9.2.

6.10.1 Unforgeability

This section shows that if the underlying DAA and ECDSA signature schemes are unforgeable and EUF-CMA secure [111] respectively, then our VDAA scheme is secure with respect to Definition 6.9.1.

Theorem 6.10.1 Let DAA be a secure DAA scheme with respect to the ideal functionality $\mathcal{F}_{\mathsf{pdaa+}}$ defined by Camenisch et al. [38] and let ECDSA be an EUF-CMA secure signature scheme, then the VDAA scheme we present in Section 6.8 is secure with respect to unforgeability as defined in Section 6.9.2.

Proof. Assume for contradiction that our VDAA scheme is not unforegeable. This means that there is an adversary $\mathcal{A}$ who manages with a non-negligible probability to win the **Forge-Game** and therefore manages to output a signature τ on a message m with respect to a pseudonym P_{ep} and a signature τ_{ep} such that $\mathsf{DSAVerify}(P_{\mathsf{ep}}, m, \tau) = \mathbf{true}$, $\mathsf{DSAVerify}(P_{\text{AA}}, P_{\mathsf{ep}}, \tau_{\mathsf{ep}}) = \mathbf{true}$ and that m does not correspond to any query made by the adversary $\mathcal{A}$ to the challenger $\mathcal{C}$.

We construct an efficient adversary $\mathcal{B}$ which uses adversary $\mathcal{A}$ to either break the unforgeability of the ideal DAA functionality $\mathcal{F}_{\mathsf{pdaa+}}$ or to win the EUF-CMA experiment.

$\mathcal{B}$ will execute $\mathcal{A}$ and simulate the challenger $\mathcal{C}$. Initially, $\mathcal{B}$ will randomly select a target vehicle $V^\star$ and a target epoch $\mathsf{ep}^\star$. The adversary $\mathcal{B}$ will simulate the Setup algorithm and will provide $\mathcal{A}$ with the resulting DAA public key ipk, the parameters par and the ECDSA public key P_{AA}. $\mathcal{B}$ also simulates N_V vehicles with identities $\{V_1, \ldots, V_{N_V}\}$. For each vehicle $\mathcal{V}_i$, $\mathcal{B}$ selects the vehicle secret key vsk_i, the Sybil secret s_i and simulates the Join protocol and the EA so that each vehicle V_i has the DAA credential cre_i on vsk_i and s_i. Finally, $\mathcal{B}$ provides $\mathcal{A}$ with a reference to each vehicle.

The adversary $\mathcal{A}$ makes a polynomial number of signature requests to the adversary $\mathcal{B}$. Each request will specify a vehicle identity $V_i \in \{V_1, \ldots, V_{N_V}\}$, a message m and an epoch ep. If $V_i = V^\star$, $\mathsf{ep} = \mathsf{ep}^\star$ and V_i does not have the signing key d_{ep} then $\mathcal{B}$ will simulate the Issue protocol with the AA using the ideal DAA functionality $\mathcal{F}_{\mathsf{pdaa}+}$ verify interface and the signature oracle $\mathcal{O}_{\mathcal{S}}$ from the EUF-CMA experiment. Once V_i has the pseudonym signing key d_{ep}, then $\mathcal{B}$ will also use the signature oracle $\mathcal{O}_{\mathcal{S}}$ from the EUF-CMA experiment to sign m.

For all other vehicles $V_i \in \{V_1, \ldots, V_{N_V}\} \setminus V^\star$, $\mathcal{B}$ will simulate V_i, will simulate the Issue protocol with the AA to generate d_{ep} if necessary and will then simulate the $\mathsf{DSASign}$ algorithm to compute the vehicle signature on m with respect to P_{ep}. In all cases, adversary $\mathcal{B}$ will provide adversary $\mathcal{A}$ with the resulting ECDSA signature τ on m, the pseudonym public key P_{ep} and the authorising AA signature τ_{ep}.

At some point $\mathcal{A}$ will terminate. By hypothesis and with a non-negligible probability $\mathcal{A}$ must output a signature τ on a message m, a pseudonym public key P_{ep} and a signature τ_{ep} such that:

- The tuple $(m^\star, \tau^\star)$ is a valid message-signature pair with respect to the pseudonym public key $P_{\mathsf{ep}}{}^\star$. In other words $\mathsf{DSAVerify}(P_{\mathsf{ep}}, m, \tau) = \mathbf{true}$.
- The message m does not correspond to any query made by the adversary $\mathcal{A}$ to adversary $\mathcal{B}$.

- The pseudonym public key and AA signature $(P_{\mathsf{ep}}{}^{\star}, \tau^{\star}{}_{\mathsf{ep}})$ is a valid message-signature pair with respect to the AA public key P_{AA}. i.e. $\mathsf{DSAVerify}(P_{\mathrm{AA}}, P_{\mathsf{ep}}{}^{\star}, \tau^{\star}{}_{\mathsf{ep}}) = \mathbf{true}$.

If $\mathcal{V}_i = \mathcal{V}^{\star}$ and $\mathsf{ep} = \mathsf{ep}^{\star}$ then $\mathcal{A}$ will send the signature τ on the message m, the pseudonym public key P_{ep} and the signature τ_{ep} to $\mathcal{B}$, otherwise it will not. This means that adversary $\mathcal{A}$ has either broken the unforgeability of the ideal DAA functionality $\mathcal{F}_{\mathsf{pdaa}+}$ or has broken the existential unforgeability of the ECDSA signature scheme.

The advantage of the adversary $\mathcal{A}$ winning the unforgeability game is therefore the probability that $\mathcal{A}$ attacks the target vehicle $V^{\star}$ during the epoch ep multiplied by the advantage of adversary $\mathcal{B}$ against the DAA and the ECDSA signature schemes. Since adversary $\mathcal{A}$ may attack either the signature τ on m or the signature τ_{ep} on P_{ep}, the advantage is further divided by two. Where N_{ep} is the number of different epochs that $\mathcal{A}$ requested signatures for and N_V is the number of vehicles simulated by $\mathcal{B}$, the advantage of $\mathcal{A}$ winning the unforgeability game is

$$\mathbf{Adv}[\mathcal{A}^{\mathbf{Forge\text{-}Game}}_{\mathsf{VDAA}}] = \frac{\mathbf{max}\Big\{\mathbf{Adv}[\mathcal{B}^{\mathbf{EUF\text{-}CMA}}_{\mathsf{ECDSA}}], \mathbf{Adv}[\mathcal{B}^{\mathcal{F}_{\mathsf{pdaa}+}}_{\mathsf{DAA}}]\Big\}}{2 * N_V \cdot N_{\mathsf{ep}}}$$

6.10.2 Unlinkability

This section shows that if the underlying DAA scheme provides unlinkability then our VDAA scheme satisfies unlinkability with respect to Definition 6.9.2. Informally, an adversary cannot distinguish between messages sent by a single vehicle during two different epochs and messages sent by two different vehicles during the same two epochs because the underlying DAA scheme has the property of strong privacy. Strong privacy guarantees, provided the vehicle host is honest, that when the AA is given two DAA signatures σ_1 and σ_2 with respect to two different basenames $\mathsf{bsn}_1 \neq \mathsf{bsn}_2$, it cannot distinguish whether both signatures were created by one vehicle or two. Strong privacy holds even when the TPM is malicious and the EA is corrupt. Because our VDAA scheme uses a DAA scheme

with strong privacy, the ECDSA pseudonyms which are requested by vehicle hosts are as unlinkable as the DAA credentials that are used to request them.

Theorem 6.10.2 Let DAA be a secure DAA scheme with respect to the ideal functionality $\mathcal{F}_{\mathsf{pdaa+}}$ defined by Camenisch et al. [38], then the VDAA scheme we present in Section 6.8 is secure with respect to unlinkability as defined in Section 6.9.2.

Proof. Assume for contradiction that our VDAA scheme is not unlinkable. This means that there is an adversary $\mathcal{A}$ who manages to win the **Priv-Game** and therefore manages to output $b' = b$ with a non-negligible advantage.

We construct an efficient adversary $\mathcal{B}$ that uses $\mathcal{A}$ to distinguish between interactions with the ideal functionality $\mathcal{F}_{\mathsf{pdaa+}}$ and the underlying DAA scheme DAA. Specifically, every time a vehicle wants to sign a message m with respect to a fresh basename bsn, $\mathcal{F}_{\mathsf{pdaa+}}$ generates a fresh group secret key isk' and then signs m using isk'. Using a fresh isk for every signature guarantees that signatures are anonymous. We use adversary $\mathcal{A}$ to distinguish from the ideal functionality $\mathcal{F}_{\mathsf{pdaa+}}$ by breaking the anonymity of the DAA signatures used to request vehicle pseudonyms.

Initially, the adversary $\mathcal{B}$ will simulate the Setup algorithm and provides adversary $\mathcal{A}$ with the resulting DAA group public key pair $\mathsf{ik} = (\mathsf{ipk}, \mathsf{isk})$, the parameters par and the AA ECDSA public key pair $\mathsf{ak} = (P_{\mathrm{AA}}, d_{\mathrm{AA}})$. $\mathcal{B}$ will also simulate two vehicles V_0 and V_1 and will simulate the Join protocol and provide $\mathcal{A}$ with the vehicle secret keys $\mathsf{vsk}_0, \mathsf{vsk}_1$ and the Sybil secrets s_0, s_1, respectively. The adversary $\mathcal{B}$ will select one target vehicle $V^\star \in \{V_0, V_1\}$.

The adversary $\mathcal{A}$ selects two distinct non-overlapping epochs $\mathsf{ep0}$ and $\mathsf{ep1}$ and submits them to adversary $\mathcal{B}$. $\mathcal{B}$ selects a bit $b \leftarrow \{0, 1\}$ and then for $V_i \in \{V_b, V_{b-1}\}$ will select the epoch keys $d_{\mathsf{ep0},i}, d_{\mathsf{ep1},i}$ and computes the public keys $P_{\mathsf{ep0},i}, P_{\mathsf{ep1},i}$. For the target vehicle $V^\star \in \{V_b, V_{b-1}\}$, $\mathcal{B}$ will interact with the ideal functionality $\mathcal{F}_{\mathsf{pdaa+}}$ sign interface to compute the DAA signatures σ_0, σ_1 on $P_{\mathsf{ep0},\star}$ and $P_{\mathsf{ep1},\star}$. For the non-target vehicle $\overline{\mathcal{V}^\star}$

the adversary $\mathcal{B}$ will compute the DAA signatures σ_0, σ_1 by simulating the first part of the standard Issue protocol. For $V_i \in \{V_b, V_{b-1}\}$, $\mathcal{B}$ will simulate the remainder of the Issue protocol and will provide the adversary $\mathcal{A}$ with the pseudonym signatures $\tau_{\mathsf{ep0},i}, \tau_{\mathsf{ep1},i}$ on the public keys $P_{\mathsf{ep0},i}, P_{\mathsf{ep1},i}$.

The adversary $\mathcal{A}$ will make a polynomial number of signature requests to the adversary $\mathcal{B}$. Each request will comprise a message m and an epoch ep. If $\mathsf{ep} \notin \{\mathsf{ep0}, \mathsf{ep1}\}$ then the challenger outputs $\perp$ and then, as per the unlinkability game, $\mathcal{B}$ will act according to the bit b:

- If $b = 0$, then $\mathcal{B}$ simulates vehicle V_0, simulates the $\mathsf{DSASign}$ algorithm and outputs the signed message (τ, m) with respect to pseudonym $P_{\mathsf{ep0},0}$.

- If $b = 1$ and $\mathsf{ep} = \mathsf{ep0}$ then $\mathcal{B}$ simulates vehicle V_0. If $\mathsf{ep} = \mathsf{ep1}$ then $\mathcal{B}$ simulates vehicle V_1. $\mathcal{B}$ outputs the signed message (τ, m) with respect to pseudonym $P_{\mathsf{ep0},0}$ or $P_{\mathsf{ep1},1}$, respectively.

At some point adversary $\mathcal{A}$ will terminate and by hypothesis will output $b' = b$ with a non-negligible advantage. Since the pseudonym keys are random bitstrings generated by the trusted host, $\mathcal{A}$ must have attacked the ideal DAA sign functionality $\mathcal{F}_{\mathsf{pdaa}+}$ used to request the pseudonym signatures. The advantage of $\mathcal{A}$ in the unlinkability game is therefore the product of the probability that the target vehicle $V^\star$ is exposed by the bit value b, the probability that $\mathcal{A}$ attacks the target vehicle $V^\star$ and the advantage of adversary $\mathcal{B}$ against the $\mathcal{F}_{\mathsf{pdaa}+}$ sign interface.

$$\mathbf{Adv}[\mathcal{A}_{\mathsf{VDAA}}^{\mathbf{Priv\text{-}Game}}] = \frac{1}{4} \cdot \mathbf{Adv}[\mathcal{B}_{\mathsf{DAA}}^{\mathcal{F}_{\mathsf{pdaa}+}}]$$

6.11 Evaluation

This section argues that the VDAA scheme we presented in Section 6.8 meets the standard security and privacy requirements for V2X from Section 6.4.

Authentication The primary security requirement for V2X is that there is a mechanism for determining the authenticity and integrity of broadcast messages. We proved that our VDAA scheme is secure with respect to unforgeability in Section 6.10.1. The unforgeability of our scheme gives an assurance that when the EA and all vehicle TPMs are honest, no adversary can forge a request for a pseudonym certificate from the AA. Correspondingly, if all TPMs and the EA are uncorrupted then no adversary can create a valid signature on any message, all messages unforgeably originate from a particular honest vehicle and the authentication and integrity of received messages is guaranteed.

Unlinkability The main privacy mechanism in V2X is the use of multiple pseudonymous identities such that an adversary is unable to distinguish whether two spatio-temporally uncorrelated identities originate from a single source or not. We proved that our VDAA scheme is secure with respect to unlinkability in Section 6.10.2. The unlinkability of our scheme assures that even if the EA is corrupt or collaborates with the AA, the signatures on broadcast messages sent by a vehicle are indistinguishable from those created by a different vehicle.

Corrupt CA Resistance Vehicles should be protected from dishonest or collaborating certificate authorities. In contrast to the leading V2X standards [88, 29] our scheme provably retains vehicle unlinkability despite dishonest certificate authorities.

Revocation It is critically important that vehicles that send false information can be prevented from continued participation. Our VDAA scheme allows both vehicle,

private-key and signature based revocation which we describe in Section 6.8. Unlike other solutions that also provide enhanced vehicle privacy [102, 256, 257], we uniquely retain centralised control over revocation and are therefore able remove vehicles despite vehicle hosts that may refuse to forward messages to the TPM.

Sybil Resistance We optimally limit Sybil attacks by restricting each vehicle to a single pseudonym request per epoch. Requests for multiple pseudonyms in the same epoch are denied, forfeit vehicle unlinkability and are detected by the AA. At most, a vehicle can use just two pseudonyms concurrently and only during the small certificate overlap period that is necessary for harmonising vehicles without a synchronised clock source.

Performance Analysis

The most performance critical operation in V2X is broadcast message signature verification. Early field studies [214] indicated that vehicles should be able to verify at least 1000 signatures per second in order to deal with busy intersections and this led to the selection of ECDSA by the major standards. Since our VDAA scheme uses regular ECDSA signatures on broadcast messages we occur no additional overhead with regards to verification speed. Similarly, the modest requirement for signing 10 messages per second poses no obstacle as we only slightly increase the complexity of the DSASign operation by two modular multiplications. In line with the standards we use either NIST curve P-256 [70] or BrainpoolP256r1 [163] which result in a broadcast message signature size of 64 bytes.

Where our scheme introduces an overhead compared to the standards is when vehicles are enrolled for the first time and, more importantly, each time they request a pseudonym certificate. The DAA credential and signature sizes used in our Join and Issue protocols depend on the underlying DAA scheme. For the q-SDH-based instantiation of Camenisch et al. [39] the DAA credential size is 96 bytes, composed of 2 elements in $\mathbb{Z}_p$ and one in $\mathbb{G}_1$.

The corresponding signature size is 356 bytes, composed from 6 elements in $\mathbb{Z}_p$, 4 elements in $\mathbb{G}_1$ and one 32 byte hash digest. The total bandwidth requirements of our Issue protocol, run each time a vehicle requests a new pseudonym, is one ECDSA pseudonym public key P_{ep} and signature τ_{ep}, one 356 byte DAA signature σ and a 4 byte epoch identifier ep. In other words, compared to the ETSI standard our scheme requires an additional 360 bytes of bandwidth per pseudonym that is requested by each vehicle.

Centralised revocation depends on the attestation list Auth-L, maintained by the AA, which retains all of the randomised DAA signature and ECDSA pseudonym tuples received during all runs of the Join protocol. Each tuple in Auth-L, comprising one 356 byte DAA signature and one 64 byte ECDSA pseudonym public key, requires 420 bytes of storage. Taking a 5 minute pseudonym validity period, the upper bound on the AA storage required is 118.125 KB per vehicle, per day. This number scales linearly in the proportion of time that a vehicle is driven for, for example reducing to less than 5 KB for vehicles used for one hour per day.

Computationally, each pseudonym request requires one DAA sign operation which takes approximately 20 ms [39] for q-SDH DAA. The DAA verification algorithm run by the AA is also efficient and takes around 60 ms.

Since our VDAA scheme authenticates each broadcast message with a standard ECDSA signature, our scheme has the same signature and certificate bandwidth overheads as the ETSI approach. Assuming the standard epoch duration of 5 minutes, and that authorising certificates are included in one out of every 10 messages sent by a vehicle, Figure 6.3 shows the bandwidth required by both our solution and ETSI's in contrast to the direct application of DAA [50]. Including the certificate in one out of every 10 CAM is a pessimistic estimate when considering that this captures the scenario in which all interactions lasting more than one second result in sending the necessary certificate.

Whilst VDAA requires no additional CAM bandwidth over the ETSI standard, a small

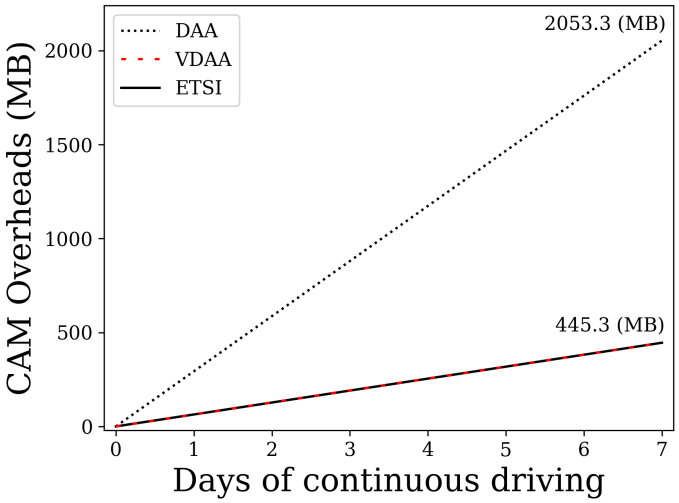


Figure 6.3: With an epoch duration of 5 minutes, a comparison of the CAM signature and certificate bandwidth overheads between the ETSI standard, our VDAA scheme and the direct application of DAA.

certificate issuance overhead is required for the DAA signatures used to request each certificate. In particular, VDAA requires the additional transmission of one 356 byte DAA signature and one 4 byte epoch identifier per epoch. The 5 minute value used in our analysis of is chosen for conformity with the value used by ETSI when evaluating different pseudonym change strategies [84] and is also the value recommended by SAE [207]. Figure 6.4 shows how the VDAA certificate issuance overhead scales to epoch durations ranging from 1 to 30 minutes. We note that even with an epoch duration of only 1 minute, VDAA requires less than 4.8 MB of vehicle-to-AA certificate issuance bandwidth per week of continuous driving. Using the recommended 5 minute epoch period, no more than 701 KB is needed per week. In practice, as most vehicles are only operated for a small proportion of each day, the overheads will be much less than the upper bounds shown here. In addition this bandwidth is only required periodically and can be scheduled according to the connectivity available to each vehicle. When considering all overheads including CAM authentication and certificate issuance in this model, VDAA requires less than 22% of the total bandwidth needed for the direct application of DAA (e.g. Chen at al. [50]).

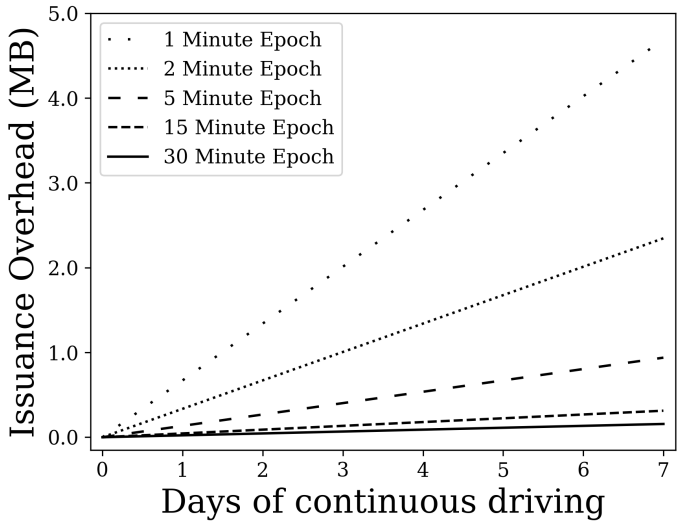


Figure 6.4: A comparison of the additional certificate issuance bandwidth required by VDAA, in contrast to the ETSI standard, at epoch durations ranging from 1 to 30 minutes.

Revocation in DAA is an inefficient process which is linear in the size of the revocation list [154]. One way of minimising the performance impact of revocation is to implement DAA groups associated with short periods of time. For example, the EA could create a DAA group for each week. Every week, each vehicle would prove that it is a non-revoked member of the current group and would be issued a new credential. The AA may even forego revocation altogether and simply wait for revoked vehicles to be removed during weekly re-keying.

6.12 Chapter Summary

In this chapter we have presented a novel V2X architecture that harmonises the strong privacy guarantees of DAA with the standard requirements for V2X introduced in Section 2.5.4. Our VDAA scheme, which we have shown secure under the standard assumptions for DAA, is compatible with the PKI architectures of the latest proposed ITS standards [88, 29] and addresses the currently unmet need [201, 183] for measures which limit long-term vehicle tracking and that minimise the impact of certificate authority collusion. Relative to the standards and many of the proposals in the literature [244, 29, 102, 50] our scheme provides a stronger security model and a higher degree of privacy. Rather than forfeiting their canonical identity, vehicles that send malicious messages or which request multiple pseudonyms for the same epoch only forfeit their unlinkability and ongoing participation in the scheme. Uniquely in the literature [102, 50, 256, 8, 208] our scheme retains centralised authority over vehicle revocation, efficient standards-compliant ECDSA signatures on broadcast messages and does not require the TPM or the certificate authorities to be trusted for privacy.

Chapter 7

Closing Remarks

In this final chapter we review and conclude the contributions we have made in this thesis and identify the directions for future research. At a high-level, we have identified several new cryptographic and key management flaws in an existing automotive immobiliser system and developed two new V2X architectures for improving the safety and privacy of tomorrow's connected and autonomous vehicles. Holistically, we have considered the history, present and future of digital automotive security and safety systems.

History

One of the most effective digital vehicle security technologies over the last 30 years is the electronic vehicle immobiliser. Electronic immobiliser systems have been mandatory in all new passenger cars sold within the EU since 1998 [55] and are estimated to have reduced the general rate of vehicle theft by 40% [243]. For many years only weak, proprietary cryptography was implemented in automotive immobiliser systems worldwide [105]. In Chapter 3 we review the academic body of work that has systematically exposed, analysed and challenged the weak and proprietary cryptography that has been the foundations of essentially all vehicle immobiliser systems worldwide [28, 247, 126, 249, 248]. Whilst research indicates [247] that the majority of vehicles sold in Europe between 1995 and 2015 are fitted with an immobiliser system based on either the Hitag2 [248] or Megamos

Crypto [238] encryption algorithm, there are still a number of widespread algorithms and systems that have received little attention. In addition, despite calls for automotive system designers to embrace industry standard cryptographic algorithms and peer-reviewed protocols that can be traced back to the at least 2005 [28], automotive manufactures have been shown to reuse old designs for new purposes, in new systems, and with little regard for the prior failings [264].

In Chapter 4 we present a thorough analysis of a popular and previously unstudied vehicle immobiliser system and algorithm, AUT64. We present full details of AUT64 including a complete specification and analysis of the proprietary block cipher, the associated authentication protocol, and its implementation in a widely-used vehicle immobiliser system that we have reverse engineered. We identify a number of cryptographic weaknesses in AUT64 and develop several attacks on both the 8 and 24 round implementations. Despite AUT64 having a 120 bit secret key length, we show that in certain implementations 8 round AUT64 can be broken within milliseconds using a standard laptop, with a worst-case complexity of $2^{37.3}$ encryptions. In the Mazda immobiliser system that we evaluate, the security of 24 round AUT64 is no more than 48.3 bits and can be exhaustively searched. We show that part of the secret key is actually derived from the public transponder identification code and can be efficiently determined from only a single interaction with the transponder. In addition to the immediate impact on automotive security, this chapter also provides a more general contribution to the literature on the cryptanalysis of generalised Feistel ciphers with key-dependent permutations and S-Boxes.

There are a number of key research directions that are motivated by the insecurity of proprietary cryptographic systems. As we have shown in Chapters 3 and 4 there is real threat that many of the systems that we rely upon to secure our vehicles [105, 28, 126, 249, 248], and more generally to protect our workplaces, public transportation systems, critical national infrastructure [104], cordless phones [181] and wireless internet access

[215], are based on weak cryptographic primitives or protocols that have not been properly audited to ensure that they provide a degree of security that is proportional to the value of what they protect.

Firstly, there is an ongoing need for research that evaluates the security of proprietary systems and algorithms, particularly those which are widely used or that protect high-value targets. As recently as late 2018, Tesla's high-end £75,000 Model S was found to be using a RKE system based on the 40 bit DST40 symmetric encryption scheme. Even in cases where standard algorithms have been used for automotive systems, insecure protocols and a lack of proper key management have rendered the overall system unfit for purpose [105]. Systems designers from all industries should be encouraged to use standardised algorithms and peer-reviewed protocols whenever possible.

Often, as illustrated in the literature and in Chapter 4, a number of individually theoretical or minor vulnerabilities can be combined to provide a practical exploitation of a real-world system. There is a need for new techniques, frameworks and standards that enable the overall security of these systems to be measured. Such research would look at developing new models which consider the properties of cryptographic primitives, the formal and symbolic verification of security protocols and the management of cryptographic keys in a holistic setting. Developing accessible tools that automate the analysis of proprietary systems and for generating secure system code from simple relational models are also important research challenges.

Finally, there is a need to design new primitives and techniques that are tailored to the specific demands of automotive systems. In particular, there are ongoing efforts [173] to understand the requirements for lightweight cryptographic primitives, to develop suitable algorithms and to produce appropriate standards. An important technique for securing vehicle RKE is RF distance bounding protocols [196], addressing the provable security of which remains an open challenge [11].

Present

The rapid digitalisation of modern vehicles has not yet been paralleled by the development of techniques or standards which adequately address the cyber security challenges posed by these systems. The current state of the art in automotive security has been described as rudimentary and "comparable to computers in the early days of the Internet" [103]. In Chapter 3 we briefly review the insecurity of intra-vehicle communication over the CAN bus and the associated research challenges. The CAN bus and the ECUs that communicate over it were not designed to be secure under the threat model of a modern vehicle in which an adversary may be able to gain access to the bus. There is a need for research that provides new methods for securing the CAN bus, for frameworks that enable different proposals to be compared and for studies that can guide the development of new interfaces and standards for secure intra-vehicle communication.

Soon, vehicles will communicate directly with surrounding vehicles and roadside infrastructure and will have advanced autonomous features which enable them to operate with little to no human input [169]. Tesla's autopilot feature is already being used on public roads in the UK [54] and Lyft are already offering rides in self-driving vehicles at select U.S. locations [165]. In the near term, a key-enabler for autonomous driving is V2X communication that will enable vehicles to develop a more detailed, contemporary and expansive model of their environment. V2X is expected to provide significant improvements in road safety and efficiency by enabling the next generation of semi-autonomous vehicle safety features such as vehicle platooning, collaborative forward collision warning and emergency electronic brake lights [246].

There are a number of key V2X standards which are supported by both European and U.S. governments [72, 53], international standardisation bodies [88, 144] and industry [156, 253]. In Chapter 5 we present IFAL, a practical and secure improvement to the leading European standard for V2X. IFAL is based on a novel key diversification mechanism

in the public-key setting that improves support for vehicles with limited bandwidth and intermittent connectivity. Each vehicle is pre-issued with a certificate file that contains a lifetime supply of inactive pseudonym certificates. The file is divided into epochs and vehicles receive small, time-delayed activation codes that enable them to use the pseudonyms in a particular epoch. By adjusting the epoch duration, vehicles with a wide range of different connectivities can be provided with the same level of cryptographic unlinkability. We show that IFAL meets the standard security and privacy requirements for V2X, is provably secure and privacy conscious in a formal setting and has favourable performance in our reference implementation. IFAL and is a good candidate for integration into the European standard.

Future

Neither of the leading standards for secure V2X communication [88, 29] provides adequate protection from certificate authorities that are dishonest or that collaborate [183]. Partially, this is due to the standardisation of ECDSA for V2X. ECDSA is well suited to V2X on current and first-generation vehicle hardware because it offers small signature sizes and low-latency message verification. However, the inability to re-randomise an ECDSA signature [162] makes it impossible to strongly protect the privacy of vehicles that request pseudonyms by repeatedly presenting a long-term ECDSA certificate. In the future, vehicles will have more reliable connectivity, increased bandwidth and access to greater computational resources that will enable the use of more modern signature schemes which provide enhanced privacy features.

In Chapter 6 we develop a new security architecture for V2X that reconciles the strong privacy guarantees of DAA with the fast verification speed, small signature size and standards compliance of ECDSA signatures for V2X broadcast messages. Our VDAA scheme, which we prove secure under the standard assumptions for DAA, is compatible

with the PKI architectures of the latest proposed V2X standards [88, 29] and uniquely addresses the challenge of preventing long-term vehicle tracking in V2X, despite corrupt and collaborating certificate authorities, whilst retaining centralised authority over vehicle revocation. In comparison to the standards and many of the proposals in the literature [244, 29, 102, 50] our scheme provides a stronger security model and a higher degree of privacy. Rather than forfeiting their canonical identity, vehicles that send malicious messages or which request multiple pseudonyms for the same epoch only forfeit their unlinkability and ongoing participation in the scheme.

There are a number of open problems and future research opportunities within the connected vehicle domain. The symbolic verification of V2X protocols has already been used to identify flaws in an existing proposal and to guide the development of an improved solution [257]. In combination with the formal security methodology that we apply in Chapters 5 and 6, the symbolic verification of security protocols should be used to guide the development of future standards for V2X. Related to the improved trade-off between privacy, trust and bandwidth that we provide with our IFAL scheme in Chapter 5 and the enhanced privacy but greater communication overheads of our VDAA architecture in Chapter 6, there is a need to develop a model and a metric that, when given the dynamic situational constraints on vehicle bandwidth and location, is able to determine the best key management strategy for a particular scenario. In general, determining the optimal strategy for pseudonym change is an open problem [189] and there is a need to develop a standardised framework for test and simulation that can be used to evaluate different approaches.

Conclusion

In this thesis we have presented several new cryptographic and key management flaws in a previously unstudied vehicle immobiliser system and we have introduced two new V2X architectures for improving the security and privacy of tomorrow's connected and autonomous vehicles. In particular, we study the AUT64 automotive block cipher and its associated authentication protocol in a real-world immobiliser system. We identify a number of cryptographic and implementational flaws which we combine to present several practical key-recovery attacks. Our work on AUT64 contributes to the body of evidence which urges the automotive industry to embrace standard cryptographic algorithms and peer-reviewed protocols when developing tomorrow's new vehicles.

Our first new V2X architecture, IFAL, is focussed on providing a practical and secure improvement to the leading European standard for V2X. Specifically IFAL introduces a new standards-compliant pseudonym issuance mechanism that eliminates the trade-off between pseudonym duration and bandwidth. We show that IFAL meets the standard security and privacy requirements for V2X, is provably secure and privacy conscious in a formal setting and has favourable performance in our reference implementation. Our second new V2X architecture, VDAA, addresses the need for new techniques that preserve vehicle privacy despite dishonest or colluding certificate authorities. Uniquely in the literature, VDAA retains centralised authority over vehicle revocation, efficient standards-compliant ECDSA signatures on broadcast messages and does not require the certificate authorities to be trusted for privacy.

Appendices

Appendix A
AUT64 Implementation Details

The key-independent compression function lookup tables that we recovered from our Mazda "Module 142" immobiliser system firmware are as follows

7D	56	99	65	8C	74	82	83
9B	92	7B	A1	AA	B0	64	CF
B9	DE	5D	ED	C8	FC	46	0B
D7	1A	3F	29	C6	38	28	47

Figure A.1: T_D key derivation table.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	2	4	6	8	A	C	E	3	1	7	5	B	9	F	D
0	3	6	5	C	F	A	9	B	8	D	E	7	4	1	2
0	4	8	C	3	7	B	F	6	2	E	A	5	1	D	9
0	5	A	F	7	2	D	8	E	B	4	1	9	C	3	6
0	6	C	A	B	D	7	1	5	3	9	F	E	8	2	4
0	7	E	9	F	8	1	6	D	A	3	4	2	5	C	B
0	8	3	B	6	E	5	D	C	4	F	7	A	2	9	1
0	9	1	8	2	B	3	A	4	D	5	C	6	F	7	E
0	A	7	D	E	4	9	3	F	5	8	2	1	B	6	C
0	B	5	E	A	1	F	4	7	C	2	9	D	6	8	3
0	C	B	7	5	9	E	2	A	6	1	D	F	3	4	8
0	D	9	4	1	C	8	5	2	F	B	6	3	E	A	7
0	E	F	1	D	3	2	C	9	7	6	8	4	A	B	5
0	F	D	2	9	6	4	B	1	E	C	3	8	7	5	A

Figure A.2: The T_{offset} compression function lookup table.

T_U								T_L							
1	0	3	2	5	4	7	6	4	5	6	7	0	1	2	3
0	1	2	3	4	5	6	7	5	4	7	6	1	0	3	2
3	2	1	0	7	6	5	4	6	7	4	5	2	3	0	1
2	3	0	1	6	7	4	5	7	6	5	4	3	2	1	0
5	4	7	6	1	0	3	2	0	1	2	3	4	5	6	7
4	5	6	7	0	1	2	3	1	0	3	2	5	4	7	6
7	6	5	4	3	2	1	0	2	3	0	1	6	7	4	5
6	7	4	5	2	3	0	1	3	2	1	0	7	6	5	4
3	2	1	0	7	6	5	4	5	4	7	6	1	0	3	2
2	3	0	1	6	7	4	5	4	5	6	7	0	1	2	3
1	0	3	2	5	4	7	6	7	6	5	4	3	2	1	0
0	1	2	3	4	5	6	7	6	7	4	5	2	3	0	1
7	6	5	4	3	2	1	0	1	0	3	2	5	4	7	6
6	7	4	5	2	3	0	1	0	1	2	3	4	5	6	7
5	4	7	6	1	0	3	2	3	2	1	0	7	6	5	4
4	5	6	7	0	1	2	3	2	3	0	1	6	7	4	5
2	3	0	1	6	7	4	5	6	7	4	5	2	3	0	1
3	2	1	0	7	6	5	4	7	6	5	4	3	2	1	0
0	1	2	3	4	5	6	7	4	5	6	7	0	1	2	3
1	0	3	2	5	4	7	6	5	4	7	6	1	0	3	2
6	7	4	5	2	3	0	1	2	3	0	1	6	7	4	5
7	6	5	4	3	2	1	0	3	2	1	0	7	6	5	4
4	5	6	7	0	1	2	3	0	1	2	3	4	5	6	7
5	4	7	6	1	0	3	2	1	0	3	2	5	4	7	6

Figure A.3: The T_U and T_L key scheduling lookup tables.

Acronyms

DES Data Encryption Standard 12

DSRC Direct Short Range Communication 28

DST Digital Signature Transponder 38

EA Enrolment Authority 35

ECDSA Elliptic Curve Digital Signature Algorithm 35

ECIES Elliptic Curve Integrated Encryption Scheme 111

ECU Electronic Control Unit 2

ETSI European Telecommunications Standards Institute 28

EU European Union 2

EUF-CMA Existential Unforgeability under Chosen-Message Attack 129

EVITA "E-safety Vehicle Intrusion Protected Applications" 30

FCC Federal Communications Commission 28

GPS Global Positioning System 50

HMAC Keyed-Hash Message Authentication Code 39

HSM Hardware Security Module 30

IDIC IDentification IC 67

IFAL Issue First Activate Later 5

ISO International Organization for Standardisation 49

ITS Intelligent Transportation Systems 27

ITS-S ITS-Stations 27

IVHS Intelligent Vehicle-Highway Systems 48

SAE Society of Automotive Engineers 53

SCMS Security Credential Management System 34

SeVeCOM Secure Vehicle COMmunication 51

SIM Subscriber Identity Module 124

SMS Short Message Service 117

SPN Substitution-Permutation Network 82

TCG Trusted Computing Group 153

TE Trusted Element 31

TPM Trusted Platform Module 153

U.S. DOT U.S. Department Of Transportation 28

UDS Unified Diagnostic Services 46

UFN Unbalanced Feistel Network 68

V2I Vehicle-to-Infrastructure 27

V2V Vehicle-to-Vehicle 27

V2X Vehicle-to-Everything 1

VANET Vehicular Ad-hoc Network 27

VDAA Vehicular DAA 151

VW Volkswagen 41

WAVE Wireless Access in Vehicular Environments 28

WEP Wired Equivalent Privacy 25

WLAN Wireless Local Area Network 28

List of References

[1] A European strategy on Cooperative Intelligent Transport Systems, a milestone towards cooperative, connected and automated mobility. *European Commission*, November 2016.

[2] Tolga Acar, Greg Zaverucha, and Lan Nguyen. A TPM Diffie-Hellman Oracle. Technical Report MSR-TR-2013-105, October 2013. URL: `https://www.microsoft.com/en-us/research/publication/a-tpm-diffie-hellman-oracle/`.

[3] ACEA Report: Vehicles in use - Europe 2018. Technical report, Association des Constructeurs Européens d'Automobiles (ACEA), January 2019. [Online, accessed 08-May-2019]. URL: `https://web.archive.org/web/20190122144916/https://www.acea.be/uploads/statistic_documents/ACEA_Report_Vehicles_in_use-Europe_2018.pdf`.

[4] Giovanni Adorni and Gianni Conte. The Connection Machine at the University of Parma. *International Journal of Modern Physics C*, 4(1):1–4, 1993. `doi:10.1142/S0129183193000021`.

[5] Advanced Encryption Standard (AES). *National Institute of Standards and Technology (NIST). U.S. Department of Commerce*, FIPS 197, November 2001.

[6] Algorithms, Key Size and Protocols Report. Technical report, D5.4, H2020-ICT-2014, Project 645421, ECRYPT-CSA, February 2019. [Online, accessed 06-June-2019]. URL: `https://web.archive.org/web/20190513122208/http://www.ecrypt.eu.org/csa/documents/D5.4-FinalAlgKeySizeProt.pdf`.

[7] ANSI X3.92. Data Encryption Algorithm. *American National Standards Institute (ANSI)*, 1981.

[8] F. Armknecht, A. Festag, D. Westhoff, and K. Zeng. Cross-layer Privacy Enhancement and Non-repudiation in Vehicular Communication. In *Communication in Distributed Systems - 15. ITG/GI Symposium*, pages 1–12, February 2007.

[9] Man Ho Au, Willy Susilo, and Yi Mu. Constant-Size Dynamic k-TAA. In Roberto De Prisco and Moti Yung, editors, *Security and Cryptography for Networks*, pages 111–125, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[10] Autonomous Vehicles Meet Human Drivers: Traffic Safety Issues for States. Technical report, Governors Highway Safety Association (GHSA), January 2017.

[11] Gildas Avoine, Muhammed Ali Bingöl, Ioana Boureanu, Srdjan čapkun, Gerhard Hancke, Süleyman Kardaş, Chong Hee Kim, Cédric Lauradoux, Benjamin Martin, Jorge Munilla, Alberto Peinado, Kasper Bonne Rasmussen, Dave Singelée, Aslan Tchamkerten, Rolando Trujillo-Rasua, and Serge Vaudenay. Security of Distance-Bounding: A Survey. *ACM Comput. Surv.*, 51(5):94:1–94:33, September 2018. URL: http://doi.acm.org/10.1145/3264628, doi:10.1145/3264628.

[12] Steve Babbage, Anne Canteaut, Carlos Cid, Henri Gilbert, Thomas Johansson, Matthew Parker, Bart Preneel, Vincent Rijmen, and Matthew Robshaw. The eSTREAM portfolio. Technical report, April 2008. [Online, accessed 07-June-2019]. URL: https://web.archive.org/web/20170705114822/http://www.ecrypt.eu.org/stream/portfolio.pdf.

[13] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK Lightweight Block Ciphers. In *Proceedings of the 52Nd Annual Design Automation Conference*, DAC '15, pages 175:1–175:6, New York, NY, USA, 2015. ACM. URL: http://doi.acm.org/10.1145/2744769.2747946, doi:10.1145/2744769.2747946.

[14] Ryad Benadjila, Mathieu Renard, José Lopes-Esteves, and Chaouki Kasmi. One Car, Two Frames: Attacks on Hitag-2 Remote Keyless Entry Systems Revisited. In *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, Vancouver, BC, 2017. USENIX Association. URL: https://www.usenix.org/conference/woot17/workshop-program/presentation/benadjila.

[15] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, January 2003. doi:10.1109/MPRV.2003.1186725.

[16] Daniel J. Bernstein, Mike Hamburg, Anna Krasnova, and Tanja Lange. Elligator: elliptic-curve points indistinguishable from uniform random strings. In *Proceedings of the 2013 ACM SIGSAC conference on computer communications security*, CCS '13, pages 967–980, New York, NY, USA, 2013. ACM. URL: http://doi.acm.org/10.1145/2508859.2516734, doi:10.1145/2508859.2516734.

[17] Eli Biham and Adi Shamir. *Differential Cryptanalysis of DES-like Cryptosystems*, pages 2–21. Springer Berlin Heidelberg, Berlin, Heidelberg, 1991. doi:10.1007/3-540-38424-3_1.

[18] Eli Biham and Adi Shamir. Differential Cryptanalysis of the Full 16-round DES. In Ernest F. Brickell, editor, *Advances in Cryptology — CRYPTO' 92*, pages 487–496, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[19] Alex Biryukov. *Weak keys*, pages 656–657. Springer US, Boston, MA, 2005. doi:10.1007/0-387-23483-7_458.

[20] Alex Biryukov and David Wagner. Slide Attacks. In Lars Knudsen, editor, *Fast Software Encryption*, pages 245–259, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[21] Alex Biryukov and David Wagner. Advanced Slide Attacks. In Bart Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, pages 589–606, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[22] Norbert Bißmeyer, S. Mauthofer, K. M. Bayarou, and F. Kargl. Assessment of node trustworthiness in VANETs using data plausibility checks with particle filters. In *2012 IEEE Vehicular Networking Conference (VNC)*, pages 78–85, November 2012. doi:10.1109/VNC.2012.6407448.

[23] Norbert Bißmeyer, Hagen Stübing, Elmar Schoch, Stefan Götz, Jan Peter Stotz, and Brigitte Lonc. A generic public key infrastructure for securing Car-to-X communication. In *18th World Congress on Intelligent Transport Systems*, October 2011.

[24] Kenneth P. Bogart. *Introductory Combinatorics*. page 486, Saunders College Publishing, Philadelphia, PA, USA, 2nd edition, 1989.

[25] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007*, pages 450–466, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[26] Andrey Bogdanov. Linear Slide Attacks on the KeeLoq Block Cipher. In Dingyi Pei, Moti Yung, Dongdai Lin, and Chuankun Wu, editors, *Information Security and Cryptology*, pages 66–80, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[27] Dan Boneh and Xavier Boyen. Short Signatures Without Random Oracles. In Christian Cachin and Jan L. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, pages 56–73, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[28] Stephen C. Bono, Matthew Green, Adam Stubblefield, Ari Juels, Aviel D. Rubin, and Michael Szydlo. Security Analysis of a Cryptographically-enabled RFID Device. In *Proceedings of the 14th Conference on USENIX Security Symposium - Volume 14*, SSYM'05, Berkeley, CA, USA, 2005. USENIX Association.

[29] B. Brecht, D. Therriault, A. Weimerskirch, W. Whyte, V. Kumar, T. Hehn, and R. Goudy. A Security Credential Management System for V2X Communications. *IEEE Transactions on Intelligent Transportation Systems*, 19(12), 2018.

[30] Ernie Brickell, Jan Camenisch, and Liqun Chen. Direct Anonymous Attestation. In *Proceedings of the 11th ACM Conference on Computer and Communications Security*, CCS '04, 2004.

[31] Ernie Brickell, Liqun Chen, and Jiangtao Li. Simplified security notions of direct anonymous attestation and a concrete scheme from pairings. *International Journal of Information Security*, 8, 2009. doi:10.1007/s10207-009-0076-3.

[32] Austin Brown, Brittany Repac, and Jeff Gonder. Autonomous Vehicles Have a Wide Range of Possible Energy Impacts (Poster). *Workshop on Road Vehicle Automation*, July 2013.

[33] Daniel R. L. Brown, Robert Gallant, and Scott A. Vanstone. Provably Secure Implicit Certificate Schemes. In Paul Syverson, editor, *Financial Cryptography*, pages 156–165, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[34] M. Bruhnke and F. Friedrich. Method of cryptological authentification in a scanning identification system, January 1997. US Patent US6510517.

[35] BSI TR-03111 Elliptic Curve Cryptography. Technical report, Bundesamt für Sicherheit in der Informationstechnik, June 2012.

[36] Douglas Busvine. Explainer: Betting on the past? Europe decides on connected car standards, April 2019. Reuters, [Online, accessed 22-May-2019]. URL: https://web.archive.org/web/20190512120734/https://reut.rs/2Ir8GOB.

[37] Levente Buttyán, Tamás Holczer, and István Vajda. On the Effectiveness of Changing Pseudonyms to Provide Location Privacy in VANETs. In Frank Stajano, Catherine Meadows, Srdjan Capkun, and Tyler Moore, editors, *Security and Privacy in Ad-hoc and Sensor Networks*, pages 129–141, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[38] J. Camenisch, L. Chen, M. Drijvers, A. Lehmann, D. Novick, and R. Urian. One TPM to Bind Them All: Fixing TPM 2.0 for Provably Secure Anonymous Attestation. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 901–920, May 2017.

[39] Jan Camenisch, Manu Drijvers, and Anja Lehmann. Anonymous Attestation Using the Strong Diffie Hellman Assumption Revisited. In Michael Franz and Panos Papadimitratos, editors, *Trust and Trustworthy Computing*, pages 1–20, Cham, 2016. Springer International Publishing.

[40] Jan Camenisch, Manu Drijvers, and Anja Lehmann. Anonymous Attestation with Subverted TPMs. In *Advances in Cryptology – CRYPTO 2017*, 2017.

[41] Jan Camenisch and Anna Lysyanskaya. A Signature Scheme with Efficient Protocols. In Stelvio Cimato, Giuseppe Persiano, and Clemente Galdi, editors, *Security in*

Communication Networks, pages 268–289, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[42] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups. In Burton S. Kaliski, editor, *Advances in Cryptology — CRYPTO '97*, pages 410–424, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[43] Ian Catling and Bob McQueen. Road transport informatics in Europe - major programs and demonstrations. *IEEE Transactions on Vehicular Technology*, 40(1):132–140, February 1991. doi:10.1109/25.69980.

[44] CEN - ENV 12253. Road Transport and Traffic Telematics (RTTT) - Dedicated Short Range Communication (DSRC) - Physical Layer Using Microwave at 5.8 GHz. Technical report, Comité Européen de Normalisation (CEN), October 1997.

[45] CEN/TC 278 Intelligent Transport Systems (ITS), August 2011. [Online, accessed 22-June-2019]. URL: https://web.archive.org/web/20190622105739/https://www.itsstandards.eu/2-uncategorised.

[46] Silvio Cesare. Breaking the Security of Physical Devices, Presentation at Black Hat USA, August 2014.

[47] David Chaum. Blind signatures for untraceable payments. In *Advances in cryptology*, pages 199–203. Springer, 1983.

[48] David L. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Commun. ACM*, 24(2):84–90, February 1981. URL: http://doi.acm.org/10.1145/358549.358563, doi:10.1145/358549.358563.

[49] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. Comprehensive Experimental Analyses of Automotive Attack Surfaces. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, Berkeley, CA, USA, 2011. USENIX Association. URL: http://dl.acm.org/citation.cfm?id=2028067.2028073.

[50] L. Chen, S. Ng, and G. Wang. Threshold Anonymous Announcement in VANETs. *IEEE Journal on Selected Areas in Communications*, 29(3):605–615, March 2011. doi:10.1109/JSAC.2011.110310.

[51] Liqun Chen and Jiangtao Li. Flexible and Scalable Digital Signatures in TPM 2.0. In *"Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security"*, CCS '13, page 37–48, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2508859.2516729.

[52] Vladimir Chepyzhov and Ben Smeets. On A Fast Correlation Attack on Certain Stream Ciphers. In Donald W. Davies, editor, *Advances in Cryptology — EUROCRYPT '91*, pages 176–185, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg.

[53] Coalition for Safety Sooner, January 2018. A letter from U.S. DOT local authorities to the Federal Communications Commission (FCC) in support of accelerated utilisation of DSRC for safety critical transportation applications, [Online, accessed 22-May-2019]. URL: `https://web.archive.org/web/20190522111307/https://news.transportation.org/Documents/spectrum%20letter.pdf`.

[54] Code of Practice: Automated vehicle trialling. Technical report, Department for Transport. UK Government, February 2019.

[55] Commission Directive 95/56/EC of 8 November 1995 adapting to technical progress Council Directive 74/61/EEC relating to devices to prevent the unauthorized use of motor vehicles. *European Commission*, December 1995.

[56] Connected and Autonomous Vehicles: Revolutionising Mobility in Society. *Society of Motor Manufacturers and Traders (SMMT)*, March 2017.

[57] Nicolas T. Courtois, Gregory V. Bard, and David Wagner. Algebraic and Slide Attacks on KeeLoq. In Kaisa Nyberg, editor, *Fast Software Encryption*, pages 97–115, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[58] Nicolas T. Courtois, Sean O'Neil, and Jean-Jacques Quisquater. Practical Algebraic Attacks on the Hitag2 Stream Cipher. In Pierangela Samarati, Moti Yung, Fabio Martinelli, and Claudio A. Ardagna, editors, *Information Security*, pages 167–176, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[59] Crypto Contactless Identification Device, October 1997. EM Microelectronic V4070 datasheet.

[60] 125kHz CRYPTO READ/WRITE Contactless Identification Device, March 2002. EM Microelectronic EM4170 datasheet.

[61] Cryptography for a Connected World, January 2011. IBM's 100 Icons of Progress, [Online, accessed 09-June-2019]. URL: `https://web.archive.org/web/20190105050439/https://www.ibm.com/ibm/history/ibm100/us/en/icons/cryptography/`.

[62] Joan Daeman and Vincent Rijmen. AES Proposal: Rijndael. *National Institute of Standards and Technology (NIST) AES Selection Process*, September 1999.

[63] Data Encryption Standard (DES). *National Institute of Standards and Technology (NIST). U.S. Department of Commerce*, FIPS 46, January 1977.

[64] Data Encryption Standard (DES). *National Institute of Standards and Technology (NIST). U.S. Department of Commerce*, FIPS 46-2, December 1993.

[65] Data Encryption Standard (DES). *National Institute of Standards and Technology (NIST). U.S. Department of Commerce*, FIPS 46-3, October 1999.

[66] Dataset: Nature of crime: vehicle-related theft. *Office for National Statistics, UK Government*, February 2019. [Online, accessed 07-May-2019]. URL: `https://www.ons.gov.uk/peoplepopulationandcommunity/crimeandjustice/datasets/natureofcrimevehiclerelatedtheft`.

[67] W. Detlefsen and W. Grabow. Interoperable 5.8 GHz DSRC Systems as Basis for Europeanwide ETC Implementation. In *1997 27th European Microwave Conference*, volume 1, pages 139–145, September 1997. `doi:10.1109/EUMA.1997.337785`.

[68] W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Trans. Inf. Theor.*, 22(6):644–654, September 2006. `doi:10.1109/TIT.1976.1055638`.

[69] W. Diffie and M. E. Hellman. Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard. *Computer*, 10(6):74–84, June 1977. `doi:10.1109/C-M.1977.217750`.

[70] Digital Signature Standard (DSS) (FIPS 186-4). Technical report, National Institute of Standards and Technology, July 2013.

[71] Itai Dinur and Adi Shamir. Cube Attacks on Tweakable Black Box Polynomials. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, pages 278–299, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[72] Directive 2010/40/EU of the European Parliament and of the Council of 7 July 2010 on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport. *Official Journal of the European Union*, August 2010.

[73] John R. Douceur. The Sybil Attack. In *Peer-to-Peer Systems*, pages 251–260, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[74] Saar Drimer and Steven J. Murdoch. Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, SS'07, pages 7:1–7:16, Berkeley, CA, USA, 2007. USENIX Association. URL: `http://dl.acm.org/citation.cfm?id=1362903.1362910`.

[75] E. Dubrova, M. Teslenko, and H. Tenhunen. On analysis and synthesis of (n,k)-non-linear feedback shift registers. In *2008 Design, Automation and Test in Europe*, pages 1286–1291, March 2008. `doi:10.1109/DATE.2008.4484856`.

[76] E-safety Vehicle Intrusion proTected Applications (EVITA) Project Summary. Technical report, April 2012. [Online, accessed 24-May-2019]. URL: https://web.archive.org/web/20170108203038/https://www.evita-project.org/Publications/EVITAD0.pdf.

[77] D. Eckhoff, C. Sommer, T. Gansen, R. German, and F. Dressler. Strong and affordable location privacy in vanets: Identity diffusion using time-slots and swapping. In *2010 IEEE Vehicular Networking Conference*, pages 174–181, December 2010. doi:10.1109/VNC.2010.5698239.

[78] Thomas Eisenbarth, Timo Kasper, Amir Moradi, Christof Paar, Mahmoud Salmasizadeh, and Mohammad T. Manzuri Shalmani. On the Power of Power Analysis in the Real World: A Complete Break of the KeeLoq Code Hopping Scheme. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, pages 203–220, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[79] Anne Ellaway, Sally Macintyre, Rosemary Hiscock, and Ade Kearns. In the driving seat: psychosocial benefits from private motor vehicle transport compared to public transport. *Transportation Research Part F: Traffic Psychology and Behaviour*, 6(3):217 – 231, 2003.

[80] ETSI EN 302 637-2. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. Technical report, V1.4.1, European Telecommunications Standards Institute, April 2019.

[81] ETSI EN 302 637-2. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service. Technical report, V1.2.2, European Telecommunications Standards Institute, November 2014.

[82] ETSI EN 302 663. Intelligent Transport Systems (ITS); ITS-G5 Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band. Technical report, European Telecommunications Standards Institute, May 2019.

[83] ETSI EN 319 411-1. Policy and security requirements for Trust Service Providers issuing certificates; Part 1: General requirements. Technical report, V1.2.0, European Telecommunications Standards Institute, August 2017.

[84] ETSI TR 103 415. Intelligent Transport Systems (ITS); Security; Pre-standardization study on pseudonym change management. Technical report, V1.1.1, European Telecommunications Standards Institute, April 2018.

[85] ETSI TS 102 731. Intelligent Transport Systems (ITS); Security; Security Services and Architecture. Technical report, V1.1.1, European Telecommunications Standards Institute, September 2010.

[86] ETSI TS 102 940. Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management. Technical report, V1.3.1, European Telecommunications Standards Institute, April 2018.

[87] ETSI TS 102 941. Intelligent Transport Systems (ITS); Security; Trust and Privacy Management. Technical report, V1.2.1, European Telecommunications Standards Institute, May 2018.

[88] ETSI TS 103 097. Intelligent Transport Systems (ITS); Security; Security header and certificate formats. Technical report, V1.3.1, European Telecommunications Standards Institute, October 2017.

[89] FCC allocates spectrum in 5.9 GHz range for intelligent transportaton systems uses. *Federal Communications Commission, U.S. Government*, October 1999. [Online, accessed 21-May-2019]. URL: `https://web.archive.org/web/20190508013952/https://transition.fcc.gov/Bureaus/Engineering_Technology/News_Releases/1999/nret9006.html`.

[90] Horst Feistel. Block cipher cryptographic system, June 1971. US Patent US3798359A.

[91] Horst Feistel. Cryptography and Computer Privacy. *Scientific American*, 228(5):15–23, 1973. URL: `http://www.jstor.org/stable/24923044`.

[92] Amos Fiat and Adi Shamir. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Advances in Cryptology — CRYPTO' 86*, 1987.

[93] Alessio Filippi, Kees Moerman, Gerardo Daalderop, Paul D. Alexander, Franz Schober, and Werner Pfliegl. Ready to roll: Why 802.11p beats LTE and 5G for V2X. Technical report, A white paper by NXP Semiconductors, Cohda Wireless, and Siemens, April 2016. [Online, accessed 22-May-2019]. URL: `https://web.archive.org/web/20190522134153/https://assets.new.siemens.com/siemens/assets/public.1510309207.ab5935c545ee430a94910921b8ec75f3c17bab6c.its-g5-ready-to-roll-en.pdf`.

[94] Alessio Filippi, Kees Moerman, Vincent Martinez, Andrew Turley, Onn Haran, and Ron Toledano. IEEE802.11p ahead of LTE-V2V for safety applications. Technical report, September 2017. NXP Semiconductors, [Online, accessed 22-May-2019]. URL: `https://web.archive.org/web/20190522140430/https://www.nxp.com/docs/en/white-paper/ROADLINK-TECH-WP.pdf`.

[95] John Flatley. Overview of vehicle-related theft: England and Wales. Technical report, Office for National Statistics. UK Government, July 2018.

[96] FMVSS No. 150 Vehicle-To-Vehicle Communication Technology For Light Vehicles. Technical report, National Highway Traffic Safety Administration (NHTSA), U.S. Department Of Transportation (USDOT), November 2016.

[97] United Nations Economic Commission for Europe. Regulation No 131 of the Economic Commission for Europe of the United Nations (UN/ECE) - Uniform provisions concerning the approval of motor vehicles with regard to the Advanced Emergency Braking Systems (AEBS). *Official Journal of the European Union*, July 2014.

[98] David Förster, Hans Löhr, Jan Zibuschka, and Frank Kargl. REWIRE – Revocation Without Resolution: A Privacy-Friendly Revocation Mechanism for Vehicular Ad-Hoc Networks", booktitle="Trust and Trustworthy Computing. pages 193–208. Springer International Publishing, 2015.

[99] Aurélien Francillon, Boris Danev, and Srdjan Capkun. Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars. In *Proceedings of the 18th annual Network and Distributed System Security (NDSS) Symposium. The Internet Society*, 2011.

[100] Robert L French, E. Ryerson Case, Yoshikazu Noguchi, Christopher Queree, Kentaro Sakamoto, and Ove Sviden. A comparison of IVHS progress in the United States, Europe, and Japan, United States Joint Program Office for Intelligent Transportation Systems, December 1993.

[101] Julien Freudiger, Mohammad Hossein Manshaei, Jean-Pierre Hubaux, and David C. Parkes. On Non-cooperative Location Privacy: A Game-theoretic Analysis. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, pages 324–337, New York, NY, USA, 2009. ACM. URL: `http://doi.acm.org/10.1145/1653662.1653702`, `doi:10.1145/1653662.1653702`.

[102] David Förster, Frank Kargl, and Hans Löhr. PUCA: A pseudonym scheme with strong privacy guarantees for vehicular ad-hoc networks. *Ad Hoc Networks*, 37:122–132, 2016. Special Issue on Advances in Vehicular Networks. `doi:https://doi.org/10.1016/j.adhoc.2015.09.011`.

[103] Flavio Garcia. Automotive Cyber Security: Lessons Learned and Research Challenges (Invited talk abstract). In *Proceedings of the Fifth Workshop on Cryptography and Security in Computing Systems*, CS2 '18, pages 19–19, New York, NY, USA, 2018. ACM. `doi:10.1145/3178291.3178295`.

[104] Flavio D. Garcia, Gerhard de Koning Gans, Ruben Muijrers, Peter van Rossum, Roel Verdult, Ronny Wichers Schreur, and Bart Jacobs. Dismantling MIFARE Classic. In S. Jajodia and J. Lopez, editors, *13th European Symposium on Research in Computer Security (ESORICS 2008)*, volume 5283 of *Lecture Notes in Computer Science*, pages 97–114. Springer Verlag, 2008.

[105] Flavio D. Garcia, David Oswald, Timo Kasper, and Pierre Pavlidès. Lock it and still lose it - on the (in)security of automotive remote keyless entry systems. In *25th*

USENIX Security Symposium (USENIX Security 2016), pages 929–944. USENIX Association, 2016.

[106] Flavio D. Garcia and Peter van Rossum. Modeling Privacy for Off-Line RFID Systems. In Dieter Gollmann, Jean-Louis Lanet, and Julien Iguchi-Cartigny, editors, *Smart Card Research and Advanced Application*, pages 194–208, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[107] O. Gehring and H. Fritz. Practical results of a longitudinal control concept for truck platooning with vehicle to vehicle communication. In *Proceedings of Conference on Intelligent Transportation Systems*, pages 117–122, November 1997. doi:10.1109/ITSC.1997.660461.

[108] Global status report on road safety 2018. Technical report, Geneva: World Health Organization, 2018.

[109] Oded Goldreich. *Foundations of Cryptography*, volume 1. Cambridge University Press, 2001. doi:10.1017/CBO9780511546891.

[110] Oded Goldreich. *Foundations of Cryptography*, volume 2. Cambridge University Press, 2004. doi:10.1017/CBO9780511721656.

[111] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-message Attacks. *SIAM J. Comput.*, 17(2):281–308, 1988. URL: http://dx.doi.org/10.1137/0217017, doi:10.1137/0217017.

[112] Jovan Dj. Golić. Cryptanalysis of Alleged A5 Stream Cipher. In Walter Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, pages 239–255, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[113] Lutz Gollan, Dr. Iur Lutz Gollan, Prof Dr, and Christoph Meinel. Digital Signatures For Automobiles?! In *Systemics, Cybernetics and Informatics (SCI)*, 2002.

[114] John Gordon, Ulrich Kaiser, and Tony Sabetti. A low cost transponder for high security vehicle immobilizers. In *29th ISATA Automotive Symposium*, pages 3–6, 1996.

[115] R.F. Graf and W. Sheets. *Video Scrambling & Descrambling: For Satellite & Cable TV*. Elsevier Science, 1998.

[116] Bogdan Groza, Stefan Murvay, Anthony van Herrewege, and Ingrid Verbauwhede. LiBrA-CAN: A Lightweight Broadcast Authentication Protocol for Controller Area Networks. In Josef Pieprzyk, Ahmad-Reza Sadeghi, and Mark Manulis, editors, *Cryptology and Network Security*, pages 185–200, Berlin, Heidelberg, 2012.

[117] Marco Gruteser and Baik Hoh. On the Anonymity of Periodic Location Samples. In Dieter Hutter and Markus Ullmann, editors, *Security in Pervasive Computing*, pages 179–192, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[118] Tim Güneysu, Timo Kasper, Martin Novotný, Christof Paar, and Andy Rupp. Cryptanalysis with COPACOBANA. *IEEE Transactions on Computers*, 57(11):1498–1513, 2008. URL: http://dx.doi.org/10.1109/TC.2008.80.

[119] G. P. Hancke. Practical attacks on proximity identification systems. In *2006 IEEE Symposium on Security and Privacy (S P'06)*, pages 6 pp.–333, May 2006. doi:10.1109/SP.2006.30.

[120] G. P. Hancke, K. E. Mayes, and K. Markantonakis. Confidence in Smart Token Proximity: Relay Attacks Revisited. *Comput. Secur.*, 28(7):615–627, October 2009. URL: http://dx.doi.org/10.1016/j.cose.2009.06.001, doi:10.1016/j.cose.2009.06.001.

[121] Ahmed Hazem and Hossam A. H. Fahmy. LCAP - A Lightweight CAN Authentication Protocol for Securing In-Vehicle Networks. In *10th International Conference on Embedded Security in Cars (ESCAR)*, Berlin, Germany, 2012.

[122] Health profile for England: 2017. Chapter 2: major causes of death and how they have changed. *Public Health England. UK Government*, July 2017. [Online, accessed 16-May-2019]. URL: https://www.gov.uk/government/publications/health-profile-for-england/chapter-2-major-causes-of-death-and-how-they-have-changed#trends-in-leading-causes-of-death.

[123] J. K. Hedrick, M. Tomizuka, and P. Varaiya. Control issues in automated highway systems. *IEEE Control Systems Magazine*, 14(6):21–32, December 1994. doi:10.1109/37.334412.

[124] Miia Hermelin and Kaisa Nyberg. Correlation properties of the bluetooth combiner. In JooSeok Song, editor, *Information Security and Cryptology - ICISC'99*, pages 17–29, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[125] Howard M. Heys. A Tutorial on Linear and Differential Cryptanalysis. *Cryptologia*, 26(3):189–221, July 2002. URL: http://dx.doi.org/10.1080/0161-110291890885, doi:10.1080/0161-110291890885.

[126] Christopher Hicks, Flavio Garcia, and David Oswald. Dismantling the AUT64 Automotive Cipher. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(2):46–69, May 2018.

[127] Christopher Hicks and Flavio D. Garcia. VDAA: A Vehicular DAA scheme for ECDSA Pseudonyms in V2X. In *IEEE European Symposium on Security and Privacy, EuroS&P*, September 2020.

[128] Philips/NXP Hitag2 PCF7936/46/47/52 stream cipher reference implementation, 2007. Anonymously published, [Online, accessed 10-June-2019]. URL: https://web.archive.org/web/20110816015908/http://cryptolib.com/ciphers/hitag2/.

[129] Tobias Hoppe, Stefan Kiltz, and Jana Dittmann. Security Threats to Automotive CAN Networks – Practical Examples and Selected Short-Term Countermeasures. In Michael D. Harrison and Mark-Alexander Sujan, editors, *Computer Safety, Reliability, and Security*, pages 235–248, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[130] IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture. *IEEE Standard 1609.0, IEEE Standards Association*, 2019.

[131] IEEE Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages. *IEEE Standard 1609.2, IEEE Standards Association*, 2016.

[132] IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services. *IEEE Standard 1609.3, IEEE Standards Association*, 2016.

[133] IEEE Standard Specifications for Public-Key Cryptography - Amendment 1: Additional Techniques. *IEEE Std 1363a-2004 (Amendment to IEEE Std 1363-2000), IEEE Standards Association*, pages 1–167, September 2004. doi:10.1109/IEEESTD.2004.94612.

[134] Vincent Immler. Breaking Hitag 2 Revisited. In Andrey Bogdanov and Somitra Sanadhya, editors, *Security, Privacy, and Applied Cryptography Engineering*, pages 126–143, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[135] Impact assessment on the Revision of the General Safety Regulation (EC) No 661/2009 and Pedestrian Safety Regulation (EC) No 78/2009. *Document 52018SC0190, European Commission*, May 2018.

[136] Sebastiaan Indesteege, Nathan Keller, Orr Dunkelman, Eli Biham, and Bart Preneel. A Practical Attack on KeeLoq. In Nigel Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, pages 1–18, Berlin, Heidelberg, 2008.

[137] Initial Cellular V2X standard completed, September 2016. [Online, accessed 22-May-2019]. URL: https://web.archive.org/web/20190522125034/https://www.3gpp.org/news-events/1798-v2x_r14.

[138] Intelligent Transport Systems in action. Technical report, Directorate-General for Mobility and Transport, European Commission, August 2010. [Online, accessed 04-June-2019]. URL: https://web.archive.org/web/20170425012120/https://www.polisnetwork.eu/uploads/Modules/PublicDocuments/intelligent-transport-systems-in-action_its-action-plan.pdf.

[139] Intelligent Vehicle-Highway Systems Act, H.R.2835, 102nd Congress, 1991. [Online, accessed 21-June-2019]. URL: https://web.archive.org/save/https://www.congress.gov/bill/102nd-congress/house-bill/2835.

[140] ISO 11898-1. Road vehicles - Controller area network (CAN) - Part 1: Data link layer and physical signalling. Technical report, International Organization for Standardization (ISO), December 2015.

[141] ISO 14229-1. Road vehicles – Unified diagnostic services (UDS) – Part 1: Specification and requirements. Technical report, International Organization for Standardization (ISO), March 2013.

[142] ISO 14230-3. Road vehicles – Diagnostic systems – Keyword Protocol 2000 – Part 3: Application layer. Technical report, International Organization for Standardization (ISO), March 1999.

[143] ISO/IEC standard 11889-1:2015: Trusted platform module library – Part 1: Architecture. Technical report, International Organization for Standardization (ISO), August 2015.

[144] ITS Standards Fact Sheets: IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE). *U.S. Department of Transportation (USDOT)*, September 2009. [Online, accessed 21-May-2019]. URL: `https://www.standards.its.dot.gov/Factsheets/PrintFactsheet/80`.

[145] Cees J. A. Jansen. *Investigations on Nonlinear Streamcipher Systems: Construction and Evaluation Methods*. Ph.D. Thesis, Technical University of Delft, 1989. URL: `http://resolver.tudelft.nl/uuid:7a13b932-d504-4fa0-866c-979daaeb4360`.

[146] Markus Kasper, Timo Kasper, Amir Moradi, and Christof Paar. Breaking KeeLoq in a Flash: On Extracting Keys at Lightning Speed. In Bart Preneel, editor, *Progress in Cryptology – AFRICACRYPT 2009*, pages 403–420, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[147] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[148] KEELOQ® Crypto Read/Write Transponder Module, 2001. HCS410/WM, Microchip Datesheet, [Online, accessed 10-June-2019]. URL: `https://web.archive.org/web/20060322005639/http://ww1.microchip.com/downloads/en/devicedoc/41116b.pdf`.

[149] Auguste Kerckhoffs. La cryptographie militaire. *Journal des Sciences*, 9(1):5–38, 1883.

[150] Sam King and Casey Henderson. Supplement to the Proceedings of the 22nd USENIX Security Symposium. In *Proceedings of the 24th USENIX Conference on Security Symposium*, SEC'15, Berkeley, CA, USA, 2015. USENIX Association.

[151] Lars Knudsen and David Wagner. *Integral Cryptanalysis*, pages 112–127. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002. doi:10.1007/3-540-45661-9_9.

[152] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental Security Analysis of a Modern Automobile. In *2010 IEEE Symposium on Security and Privacy*, pages 447–462, May 2010. doi:10.1109/SP.2010.34.

[153] Hugo Krawczyk. Cryptographic Extraction and Key Derivation: The HKDF Scheme. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, pages 631–648, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[154] Vireshwar Kumar, He Li, Noah Luther, Pranav Asokan, Jung-Min (Jerry) Park, Kaigui Bian, Martin B. H. Weiss, and Taieb Znati. Direct Anonymous Attestation with Efficient Verifier-Local Revocation for Subscription System. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ASIACCS '18, 2018.

[155] Ulf E. Larson and Dennis K. Nilsson. Securing Vehicles Against Cyber Attacks. In *Proceedings of the 4th Annual Workshop on Cyber Security and Information Intelligence Research: Developing Strategies to Meet the Cyber Security and Information Intelligence Challenges Ahead*, CSIIRW '08, pages 30:1–30:3, New York, NY, USA, 2008. ACM. URL: http://doi.acm.org/10.1145/1413140.1413174, doi:10.1145/1413140.1413174.

[156] Launch of DSRC Technology to Connect Vehicles and Infrastructure in the U.S., May 2018. CAR 2 CAR Communication Consortium Statement, [Online, accessed 22-May-2019]. URL: https://web.archive.org/web/20190522113611/https://www.car-2-car.org/fileadmin/press/pdf/CAR_2_CAR_Communication_Consortium_Press_Information_DSRC_Launch_in_USA.pdf.

[157] S. Lefèvre, J. Petit, R. Bajcsy, C. Laugier, and F. Kargl. Impact of V2X privacy strategies on Intersection Collision Avoidance systems. In *2013 IEEE Vehicular Networking Conference*, pages 71–78, December 2013. doi:10.1109/VNC.2013.6737592.

[158] Tim Leinmueller, Levente Buttyan, Jean-Pierre Hubaux, Frank Kargl, Rainer Kroh, Panagiotis Papadimitratos, Maxim Raya, and Elmar Schoch. SEVECOM - Secure Vehicle Communication. *IST Mobile and Wireless Communication Summit*, June 2006. URL: http://infoscience.epfl.ch/record/124920.

[159] T. Leinmuller, E. Schoch, and C. Maihofer. Security requirements and solution concepts in vehicular ad hoc networks. In *2007 Fourth Annual Conference on Wireless on Demand Network Systems and Services*, pages 84–91, January 2007. doi:10.1109/WONS.2007.340489.

[160] Leping Huang, K. Matsuura, H. Yamane, and K. Sezaki. Enhancing wireless location privacy using silent period. In *IEEE Wireless Communications and Networking Conference, 2005*, volume 2, pages 1187–1192, March 2005. doi:10.1109/WCNC.2005.1424677.

[161] Mingyan Li, Krishna Sampigethaya, Leping Huang, and Radha Poovendran. Swing & Swap: User-centric Approaches Towards Maximizing Location Privacy. In *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, WPES '06, pages 19–28, New York, NY, USA, 2006. ACM. URL: http://doi.acm.org/10.1145/1179601.1179605, doi:10.1145/1179601.1179605.

[162] Yehuda Lindell. Fast Secure Two-Party ECDSA Signing. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 613–644, Cham, 2017. Springer International Publishing.

[163] M. Lochter and J. Merkle. Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation. Technical report, March 2010. [Online, accessed 11-July-2019]. URL: https://web.archive.org/web/20190711055855/https://tools.ietf.org/html/rfc5639, doi:10.17487/RFC5639.

[164] Jiqiang Lu. Related-key rectangle attack on 36 rounds of the XTEA block cipher. *International Journal of Information Security*, 8(1):1–11, February 2009. doi:10.1007/s10207-008-0059-9.

[165] Lyft, Inc. *U.S. Securities and Exchange Commission*, Form S-1 Registration statement under the securities act of 1933, March 2019.

[166] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym Systems. In Howard Heys and Carlisle Adams, editors, *Selected Areas in Cryptography*, pages 184–199, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[167] Z. Ma, F. Kargl, and M. Weber. Measuring location privacy in V2X communication systems with accumulated information. In *2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*, pages 322–331, October 2009. doi:10.1109/MOBHOC.2009.5336983.

[168] Aleksandr Malchik. An English translation of [266], March 2005. A draft translation of GOST 28147-89, [Online, accessed 09-June-2019]. URL: https://web.archive.org/web/20161225015542/http://www.autochthonous.org/crypto/gosthash.tar.gz.

[169] Market Forecast for Connected and Autonomous Vehicles. Technical report, Centre for Connected and Autonomous Vehicles. UK Government, July 2017.

[170] Keith M. Martin. *Everyday Cryptography : Fundamental Principles and Applications*. Oxford University Press, 2012.

[171] Mitsuru Matsui. Linear Cryptanalysis Method for DES Cipher. In Tor Helleseth, editor, *Advances in Cryptology — EUROCRYPT '93*, pages 386–397, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[172] Mitsuru Matsui. New block encryption algorithm MISTY. In Eli Biham, editor, *Fast Software Encryption*, pages 54–68, Berlin, Heidelberg, 1997.

[173] Kerry McKay, Lawrence Bassham, Meltem Sönmez Turan, and Nicky Mouha. Report on lightweight cryptography. Technical report, NISTIR 8114, National Institute of Standards and Technology (NIST). U.S. Department of Commerce, March 2017. doi:10.6028/NIST.IR.8114.

[174] David C. McNett. US Government's Encryption Standard Broken in Less Than a Day. January 1999. [Online, accessed 15 May 2019]. URL: https://www.distributed.net/.

[175] Ralph C. Merkle. Fast Software Encryption Functions. In Alfred J. Menezes and Scott A. Vanstone, editors, *Advances in Cryptology-CRYPTO' 90*, pages 477–501, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg.

[176] Charlie Miller and Chris Valasek. Remote Exploitation Of An Unaltered Passenger Vehicle, Presentation at Black Hat USA, August 2015. [Online, accessed 12-June-2019]. URL: https://web.archive.org/web/20190416221042/http://illmatics.com/Remote%20Car%20Hacking.pdf.

[177] Nick Morgan, Oliver Shaw, Andy Feist, and Christos Byron. Reducing criminal opportunity: vehicle security and vehicle crime. *Home Office. UK Government*, Research Report 87, January 2016.

[178] Roger M. Needham and Michael D. Schroeder. Using Encryption for Authentication in Large Networks of Computers. *Commun. ACM*, 21(12):993–999, December 1978. doi:10.1145/359657.359659.

[179] Roger M Needham and David J Wheeler. Tea extensions. 1997.

[180] Karsten Nohl, David Evans, Starbug Starbug, and Henryk Plötz. Reverse-engineering a cryptographic rfid tag. In *Proceedings of the 17th Conference on Security Symposium*, SS'08, pages 185–193, Berkeley, CA, USA, 2008. USENIX Association. URL: http://dl.acm.org/citation.cfm?id=1496711.1496724.

[181] Karsten Nohl, Erik Tews, and Ralf-Philipp Weinmann. Cryptanalysis of the DECT Standard Cipher. In Seokhie Hong and Tetsu Iwata, editors, *Fast Software Encryption*, pages 1–18, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[182] Older Drivers 2015. Technical report, European Road Safety Observatory (ERSO). European Commission, 2015.

[183] Opinion 03/2017 on Processing personal data in the context of Cooperative Intelligent Transport Systems (C-ITS). Technical report, October 2017.

[184] C. Paar, T. Eisenbarth, M. Kasper, T. Kasper, and A. Moradi. KeeLoq and Side-Channel Analysis-Evolution of an Attack. In *2009 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pages 65–69, September 2009. doi:10.1109/FDTC.2009.44.

[185] Christof Paar and Jan Pelzl. *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[186] Y. Pan, J. Li, L. Feng, and B. Xu. An Analytical Model for Random Changing Pseudonyms Scheme in VANETs. In *2011 International Conference on Network Computing and Information Security*, volume 2, pages 141–145, May 2011. doi:10.1109/NCIS.2011.127.

[187] Yuanyuan Pan and Jianqing Li. Cooperative pseudonym change scheme based on the number of neighbors in VANETs. *Journal of Network and Computer Applications*, 36(6):1599–1609, 2013. doi:https://doi.org/10.1016/j.jnca.2013.02.003.

[188] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J. Hubaux. Secure vehicular communication systems: design and architecture. *IEEE Communications Magazine*, 46(11):100–109, November 2008. doi:10.1109/MCOM.2008.4689252.

[189] J. Petit, F. Schaub, M. Feiri, and F. Kargl. Pseudonym Schemes in Vehicular Networks: A Survey. *IEEE Communications Surveys Tutorials*, 17(1):228–255, 2015. doi:10.1109/COMST.2014.2345420.

[190] T. Pornin. RFC6979 - Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA). Technical report, August 2013. [Online, accessed 31-May-2019]. URL: https://web.archive.org/web/20161228222511/http://www.ietf.org/rfc/rfc6979.txt.

[191] Programme for a european traffic system with highest efficiency and unprecedented safety. PROMETHEUS Project Summary, EUREKA network [Online, accessed 03-May-2019]. URL: https://web.archive.org/web/20181116082249/https://www.eurekanetwork.org/project/id/45.

[192] Public Law 102-240: Intermodal Surface Transportation Efficiency Act of 1991, 105 Stat. 1914, 1991.

[193] Secure Hash Standard (SHS). *National Institute of Standards and Technology (NIST). U.S. Department of Commerce*, Publication 180-4, August 2015.

[194] P Rabbitt, A CARMICHAEL, S Jones, and C Holland. When and Why Older Drivers Give Up Driving. *AA Foundation for Road Safety Research*, 1996.

[195] Andreea-Ina Radu and Flavio D. Garcia. LeiA: A lightweight authentication protocol for CAN. In *21st European Symposium on Research in Computer Security (ESORICS 2016)*, volume 9879 of *Lecture Notes in Computer Science*, pages 283–300. Springer-Verlag, 2016.

[196] Kasper Bonne Rasmussen and Srdjan Čapkun. Realization of RF Distance Bounding. In *Proceedings of the 19th USENIX Conference on Security*, USENIX Security'10, pages 25–25, Berkeley, CA, USA, 2010. USENIX Association. URL: `http://dl.acm.org/citation.cfm?id=1929820.1929854`.

[197] M. Raya, P. Papadimitratos, and J. Hubaux. Securing Vehicular Communications. *IEEE Wireless Communications*, 13(5):8–15, October 2006. `doi:10.1109/WC-M.2006.250352`.

[198] Maxim Raya and Jean-Pierre Hubaux. The Security of Vehicular Ad Hoc Networks. In *Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks*, SASN '05, pages 11–21, New York, NY, USA, 2005. ACM. URL: `http://doi.acm.org/10.1145/1102219.1102223`, `doi:10.1145/1102219.1102223`.

[199] Read/Write Crypto Transponder for Short Cycle Time, Atmel TK5561A-PP datasheet, Version 4682D-RFID-09/06, September 2006.

[200] Regulation (EU) 2015/758 of the European Parliament and of the Council of 29 April 2015 concerning type-approval requirements for the deployment of the eCall in-vehicle system based on the 112 service and amending Directive 2007/46/EC. *Official Journal of the European Union*, L123/77, May 2015. URL: `https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32015R0758`.

[201] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union*, L119:1–88, May 2016. URL: `http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2016:119:TOC`.

[202] D. B. Reid. An algorithm for tracking multiple targets. In *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, pages 1202–1211, January 1978. `doi:10.1109/CDC.1978.268125`.

[203] *Reports and deliverables from the DRIVE programme, DRIVE (1989-91): Dedicated road infrastructure for vehicle safety in Europe.* Commission of the European Communities, 1991. [Online, accessed 21-June-2019]. URL: `https://web.archive.org/web/20190621133459/http://aei.pitt.edu/41466/`.

[204] *Resources for Tomorrow's Transport.* Proceedings of the Eleventh International Symposium on Theory and Practice in Transport Economics. OECD Publishing, Page 457, 1989.

[205] Ronald L. Rivest. The RC5 encryption algorithm. In Bart Preneel, editor, *Fast Software Encryption*, pages 86–96, Berlin, Heidelberg, 1995.

[206] IEEE Std. 802.11p. IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications; Amendment 6: Wireless Access in Vehicular Environments. Technical report, Institute of Electrical and Electronics Engineers (IEEE), July 2010.

[207] SAE J2735 BSM. Dedicated Short Range Communications (DSRC) Message Set Dictionary. Technical report, J2735_201603, Society of Automotive Engineers (SAE) International, March 2016.

[208] F. Schaub, F. Kargl, Z. Ma, and M. Weber. V-Tokens for Conditional Pseudonymity in VANETs. In *2010 IEEE Wireless Communication and Networking Conference*, pages 1–6, April 2010. doi:10.1109/WCNC.2010.5506126.

[209] Bruce Schneier. Description of a new variable-length key, 64-bit block cipher (Blowfish). In Ross Anderson, editor, *Fast Software Encryption*, pages 191–204, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[210] Bruce Schneier. *Applied Cryptography (2Nd Ed.): Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., New York, NY, USA, 1995.

[211] Bruce Schneier and John Kelsey. *Unbalanced Feistel networks and block cipher design*, pages 121–144. Springer Berlin Heidelberg, Berlin, Heidelberg, 1996. doi:10.1007/3-540-60865-6_49.

[212] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO*, 1989.

[213] Security Architecture and Mechanisms for V2V/V2I. Technical report, Deliverable 2.1, SEcure VEhicle COMmunication (SEVECOM) Project, February 2008. [Online, accessed 05-July-2019]. URL: https://web.archive.org/web/20190705112315/https://www.sevecom.eu/Deliverables/Sevecom_Deliverable_D2.1_v3.0.pdf.

[214] Security Requirements of Vehicle Security Architecture. Technical report, PREparing SEcuRe VEhicle-to-X Communication Systems (PRESERVE), July 2011.

[215] Pouyan Sepehrdad, Petr Sušil, Serge Vaudenay, and Martin Vuagnoux. Smashing WEP in a Passive Attack. In Shiho Moriai, editor, *Fast Software Encryption*, pages 155–178, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[216] C. E. Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28:656–715, Oct 1949. doi:10.1002/j.1538-7305.1949.tb00928.x.

[217] Akihiro Shimizu and Shoji Miyaguchi. Fast Data Encipherment Algorithm FEAL. In David Chaum and Wyn L. Price, editors, *Advances in Cryptology — EUROCRYPT' 87*, pages 267–278, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.

[218] R. Shirey. RFC4949 - Internet Security Glossary, Version 2. Technical report, August 2007. [Online, accessed 14-June-2019]. URL: `https://web.archive.org/web/20190605165509/https://tools.ietf.org/html/rfc4949`.

[219] SMMT Motor Industry Facts 2018. *The Society of Motor Manufacturers and Traders*, June 2018.

[220] Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT Solvers to Cryptographic Problems. In Oliver Kullmann, editor, *Theory and Applications of Satisfiability Testing - SAT 2009*, pages 244–257, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[221] E. Sortomme and M. A. El-Sharkawi. Optimal Scheduling of Vehicle-to-Grid Energy and Ancillary Services. *IEEE Transactions on Smart Grid*, 3(1):351–359, March 2012. `doi:10.1109/TSG.2011.2164099`.

[222] Special Publication 800-108. Recommendation for Key Derivation Using Pseudorandom Functions (Revised). *National Institute of Standards and Technology (NIST). U.S. Department of Commerce*, 2009.

[223] Special Publication 800-38B: Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication. Technical report, National Institute of Standards and Technology (NIST). U.S. Department of Commerce, June 2016.

[224] Standard Read/Write Crypto Identification IC, Atmel e5561 datasheet, Version 4699D–RFID–09/06, September 2006.

[225] Jennifer G. Steiner, Clifford Neuman, and Jeffrey I. Schiller. Kerberos: An Authentication Service for Open Network Systems. In *USENIX Conference Proceedings*, pages 191–202, 1988.

[226] Adam Stubblefield, John Ioannidis, and Aviel D. Rubin. A Key Recovery Attack on the 802.11B Wired Equivalent Privacy Protocol (WEP). *ACM Trans. Inf. Syst. Secur.*, 7(2):319–332, May 2004. `doi:10.1145/996943.996948`.

[227] Siwei Sun, Lei Hu, Yonghong Xie, and Xiangyong Zeng. Cube Cryptanalysis of Hitag2 Stream Cipher. In Dongdai Lin, Gene Tsudik, and Xiaoyun Wang, editors, *Cryptology and Network Security*, pages 15–25, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[228] Joseph M. Sussman. *ITS: A Short History and a Perspective on the Future*, pages 3–17. Springer US, Boston, MA, 2005. `doi:10.1007/0-387-23260-5_1`.

[229] P. Syverson. A taxonomy of replay attacks. In *Proceedings The Computer Security Foundations Workshop VII*, pages 187–191, June 1994. doi:10.1109/CSFW.1994.315935.

[230] Police recorded crime open data Police Force Area tables from year ending March 2013 onwards. *Home Office, UK Government*, April 2019. [Online, accessed 07-May-2019]. URL: https://www.gov.uk/government/statistics/police-recorded-crime-open-data-tables.

[231] Table VEH0101. Licensed vehicles by body type (quarterly): Great Britain and United Kingdom. *Department for Transport statistics, UK Government*, 2019. [Online, accessed 03-May-2019]. URL: https://www.gov.uk/government/statistical-data-sets/all-vehicles-veh01.

[232] Table VEH0150. Vehicles registered for the first time by body type, monthly: Great Britain and United Kingdom. *Department for Transport statistics, UK Government*, 2019. [Online, accessed 03-May-2019]. URL: https://www.gov.uk/government/statistical-data-sets/all-vehicles-veh01.

[233] S. Takaba. Japanese projects on automobile information and communication systems - Things aimed at and obtained in 20 years' experiences. In *Vehicle Navigation and Information Systems Conference, 1991*, volume 2, pages 233–240, October 1991. doi:10.1109/VNIS.1991.205768.

[234] Erik Tews, Ralf-Philipp Weinmann, and Andrei Pyshkin. Breaking 104 Bit WEP in Less Than 60 Seconds. In Sehun Kim, Moti Yung, and Hyung-Woo Lee, editors, *Information Security Applications*, pages 188–202, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[235] The Keyed-Hash Message Authentication Code (HMAC). *National Institute of Standards and Technology (NIST). U.S. Department of Commerce*, Publication 198-1, July 2008.

[236] K. Tokuda, M. Akiyama, and H. Fujii. DOLPHIN for inter-vehicle communications system. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511)*, pages 504–509, October 2000. doi:10.1109/IVS.2000.898395.

[237] Trusted Platform Module Library Part 3: Commands. Technical report, Family 2.0, Revision 01.38, Trusted Computing Group, 2016. [Online, accessed 07-June-2019]. URL: https://web.archive.org/web/20190607073909/https://www.trustedcomputinggroup.org/wp-content/uploads/TPM-Rev-2.0-Part-3-Commands-01.38.pdf.

[238] Sadayuki Tsugawa, Shin Kato, Kiyohito Tokuda, Takeshi Matsui, and Hayato Fujii. DEMO 2000 Cooperative Driving: An Overview. In *8th World Congress on Intelligent Transport Systems*, 2001.

[239] Andrew Turley, Kees Moerman, Alessio Filippi, and Vincent Martinez. C-ITS: Three observations on LTE-V2X and ETSI ITS-G5—A comparison, March 2018. NXP Semiconductor, [Online, accessed 22-May-2019]. URL: https://web.archive.org/web/20190522132723/https://www.nxp.com/docs/en/white-paper/CITSCOMPWP.pdf.

[240] Chris Valasek and Charlie Miller. Adventures in Automotive Networks and Control Units. Technical report, IOActive, 2014. [Online, accessed 14-June-2019]. URL: https://web.archive.org/web/20190319005717/https://ioactive.com/pdfs/IOActive_Adventures_in_Automotive_Networks_and_Control_Units.pdf.

[241] Jan Van den Herrewegen and Flavio D. Garcia. Beneath the bonnet: A breakdown of diagnostic security. In Javier Lopez, Jianying Zhou, and Miguel Soriano, editors, *Computer Security*, pages 305–324, Cham, 2018. Springer International Publishing.

[242] Anthony Van Herrewege, Dave Singelée, and Ingrid Verbauwhede. CANAuth - A Simple, Backward Compatible Broadcast Authentication Protocol for CAN bus. In *ECRYPT Workshop on Lightweight Cryptography*, January 2011.

[243] Jan C. van Ours and Ben Vollaard. The Engine Immobiliser: A Non-starter for Car Thieves. *The Economic Journal*, 126(593):1264–1291, 04 2015.

[244] VANETS Security Requirements Final Version. Technical report, Deliverable 1.1, SEcure VEhicle COMmunication (SEVECOM) Project, November 2006. [Online, accessed 03-July-2019]. URL: https://web.archive.org/web/20190703105529/https://www.sevecom.eu/Deliverables/Sevecom_Deliverable_D1.1_v2.0.pdf.

[245] Serge Vaudenay. The security of dsa and ecdsa. In Yvo G. Desmedt, editor, *Public Key Cryptography — PKC 2003*, pages 309–323, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[246] Vehicle-to-Vehicle Communications: Readiness of V2V Technology for Application. Technical report, DOT HS 812 014, National Highway Traffic Safety Administration (NHTSA), U.S. Department of Transportation, August 2014.

[247] Roel Verdult. *The (in)security of proprietary cryptography.* April 2015. Ph.D. Thesis, Radboud Universiteit Nijmegen. URL: https://web.archive.org/web/20170329081200/http://www.cs.ru.nl/~rverdult/phd_thesis-roel_verdult.pdf.

[248] Roel Verdult, Flavio D. Garcia, and Josep Balasch. Gone in 360 seconds: Hijacking with Hitag2. In *21st USENIX Security Symposium (USENIX Security 2012)*, pages 237–252. USENIX Association, 2012.

[249] Roel Verdult, Flavio D. Garcia, and Barış Ege. Dismantling Megamos Crypto: Wirelessly lockpicking a vehicle immobilizer. In *22nd USENIX Security Symposium (USENIX Security 2013)*, pages 703–718. USENIX Association, 2015.

[250] Eric Verheul, Christopher Hicks, and Flavio D. Garcia. IFAL: Issue First Activate Later Certificates for V2X. In *IEEE European Symposium on Security and Privacy, EuroS&P*, June 2019.

[251] Aram Verstegen, Roel Verdult, and Wouter Bokslag. Hitag 2 Hell – Brutally Optimizing Guess-and-Determine Attacks. In *12th USENIX Workshop on Offensive Technologies (WOOT 18)*, Baltimore, MD, 2018. USENIX Association. URL: `https://www.usenix.org/conference/woot18/presentation/verstegen`.

[252] Volkswagen Aktiengesellschaft v Garcia & Ors, Case No. HC13C02168, Application for an interim injunction pending trial, England and Wales High Court (Chancery Division), June 2013. [Online, accessed 11-June-2019]. URL: `https://web.archive.org/web/20170914124608/https://www.bailii.org/ew/cases/EWHC/Ch/2013/1832.html`.

[253] Volkswagen Group assumes pioneering role in rapid road safety improvement, February 2018. Volkswagen Group News, [Online, accessed 22-May-2019]. URL: `https://web.archive.org/web/20181224135516/https://www.volkswagen-newsroom.com/en/press-releases/volkswagen-group-assumes-pioneering-role-in-rapid-road-safety-improvement-541`.

[254] J. W. Wedel, B. Schünemann, and I. Radusch. V2X-Based Traffic Congestion Recognition and Avoidance. In *2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, pages 637–641, December 2009. `doi:10.1109/I-SPAN.2009.71`.

[255] David J. Wheeler and Roger M. Needham. TEA, a tiny encryption algorithm. In Bart Preneel, editor, *Fast Software Encryption*, pages 363–366, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[256] J. Whitefield, L. Chen, T. Giannetsos, S. Schneider, and H. Treharne. Privacy-enhanced capabilities for VANETs using direct anonymous attestation. In *2017 IEEE Vehicular Networking Conference (VNC)*, pages 123–130, November 2017. `doi:10.1109/VNC.2017.8275615`.

[257] Jorden Whitefield, Liqun Chen, Frank Kargl, Andrew Paverd, Steve Schneider, Helen Treharne, and Stephan Wesemeyer. Formal Analysis of V2X Revocation Protocols. In Giovanni Livraga and Chris Mitchell, editors, *Security and Trust Management*, pages 147–163, Cham, 2017. Springer International Publishing.

[258] W. Whyte, A. Weimerskirch, V. Kumar, and T. Hehn. A security credential management system for V2V communications. In *2013 IEEE Vehicular Networking Conference*, pages 1–8, December 2013. `doi:10.1109/VNC.2013.6737583`.

[259] B. Wiedersheim, Z. Ma, F. Kargl, and P. Papadimitratos. Privacy in inter-vehicular networks: Why simple pseudonym change is not enough. In *2010 Seventh International Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 176–183, February 2010. doi:10.1109/WONS.2010.5437115.

[260] M. Williams. Prometheus – the european research programme for optimising the road transport system in europe. In *IEE Colloquium on Driver Information*, pages 1–9, December 1988.

[261] M. Williams. PROMETHEUS is rolling. In *IEE Colloquium on Car and its Environment - What DRIVE and PROMETHEUS Have to Offer*, pages 1–2, January 1990.

[262] Marko Wolf, André Weimerskirch, and Thomas Wollinger. State of the Art: Embedding Security in Vehicles. *EURASIP Journal on Embedded Systems*, June 2007. doi:10.1155/2007/74706.

[263] Marko Wolf, André Weimerskirch, and Christof Paar. Security in automotive bus systems. In *Proceedings of the Workshop on Embedded Security in Cars (ESCAR)*, 2004.

[264] Lennert Wouters, Eduard Marin, Tomer Ashur, Benedikt Gierlichs, and Bart Preneel. Fast, Furious and Insecure: Passive Keyless Entry and Start Systems in Modern Supercars. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(3):66–85, May 2019. doi:10.13154/tches.v2019.i3.66-85.

[265] R. Yuan. North American dedicated short range communications (DSRC) standards. In *Proceedings of Conference on Intelligent Transportation Systems*, pages 537–542, November 1997. doi:10.1109/ITSC.1997.660531.

[266] I. A. Zabotin, G. P. Glazkov, and V. B. Isaeva. Cryptographic Protection for Information Processing Systems, Government Standard of the USSR, GOST 28147-89. Technical report, Government Committee of the USSR for Standards, 1989. In Russian, translated to English in [168].

[267] Michal Zalewski. Strange Attractors and TCP/IP Sequence Number Analysis. Technical report, 2001. [Online, accessed 12-June-2019]. URL: http://lcamtuf.coredump.cx/newtcp/.

[268] Magda El Zarki, Sharad Mehrotra, Gene Tsudik, and Nalini Venkatasubramanian. Security issues in a future vehicular network. In *European Wireless*, pages 270–274, 2002.

[269] Ke Zeng. Pseudonymous PKI for Ubiquitous Computing. In Andrea S. Atzeni and Antonio Lioy, editors, *Public Key Infrastructure*, pages 207–222, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[270] P. Štembera and M. Novotny. Breaking Hitag2 with Reconfigurable Hardware. In *2011 14th Euromicro Conference on Digital System Design*, pages 558–563, Aug 2011. doi:10.1109/DSD.2011.77.