Transmitter Authentication in Dynamic Spectrum Sharing

Vireshwar Kumar

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Engineering

Jung-Min (Jerry) Park, Chair
Thomas Hou
Michael S. Hsiao
Danfeng Yao
Chao Wang

December 12, 2016
Blacksburg, Virginia

Keywords: Dynamic spectrum sharing, spectrum security and enforcement, privacy-preserving authentication, anonymous attestation, blind authentication.

Transmitter Authentication in Dynamic Spectrum Sharing

Vireshwar Kumar

ABSTRACT

Recent advances in spectrum access technologies, such as software-defined radios, have made dynamic spectrum sharing (DSS) a viable option for addressing the spectrum shortage problem. However, these advances have also contributed to the increased possibility of "rogue" transmitter radios which may cause significant interference to other radios in DSS. One approach for countering such threats is to employ a *transmitter authentication* scheme at the physical (PHY) layer. In PHY-layer authentication, an authentication signal is generated by the transmitter, and embedded into the message signal. This enables a regulatory enforcement entity to extract the authentication signal from the received signal, uniquely identify a transmitter, and collect verifiable evidence of a rogue transmission that can be used later during an adjudication process. There are two primary technical challenges in devising a transmitter authentication scheme for DSS: (1) how to generate and verify the authentication signal such that the required security and privacy criteria are met; and (2) how to embed and extract the authentication signal without negatively impacting the performance of the transmitters and the receivers in DSS. In this dissertation, with regard to dealing with the first challenge, the novel approaches which significantly improve scalability of the transmitter authentication with respect to revocation in large networks, are proposed. With regard to dealing with the second challenge, the novel approaches which are not constrained by the tradeoff between the message signal's signal to interference and noise ratio (SINR) and the authentication signal's SINR, are proposed.

Transmitter Authentication in Dynamic Spectrum Sharing

Vireshwar Kumar

GENERAL AUDIENCE ABSTRACT

Recent advances in spectrum access technologies, such as software-defined radios, have made dynamic spectrum sharing (DSS) a viable option for addressing the spectrum shortage problem. However, these advances have also contributed to the increased possibility of "rogue" transmitter radios which may cause significant interference to other radios in DSS. One approach for countering such threats is to employ a *transmitter authentication* scheme at the physical (PHY) layer. In PHY-layer authentication, an authentication signal is generated by the transmitter, and embedded into the message signal. This enables a regulatory enforcement entity to extract the authentication signal from the received signal, uniquely identify a transmitter, and collect verifiable evidence of a rogue transmission that can be used later during an adjudication process. There are two primary technical challenges in devising a transmitter authentication scheme for DSS: (1) how to generate and verify the authentication signal such that the required security and privacy criteria are met; and (2) how to embed and extract the authentication signal without negatively impacting the performance of the transmitters and the receivers in DSS. With regard to dealing with the first challenge, the authentication schemes in the prior art, which provide privacy-preserving authentication, have limited practical value for use in large networks due to the high computational complexity of their revocation check procedures. In this dissertation, the novel approaches which significantly improve scalability of the transmitter authentication with respect to revocation, are proposed. With regard to dealing with the second challenge, in the existing PHY-layer authentication techniques, the authentication signal is embedded into the message signal in such a way that the authentication signal appears as noise to the message signal and vice versa. Hence, existing schemes are constrained by a fundamental tradeoff between the message signal's signal to interference and noise ratio (SINR) and the authentication signal's SINR. In this dissertation, the novel approaches which are not constrained by the aforementioned tradeoff between message and authentication signals, are proposed.

Dedication

I dedicate this dissertation to my loving parents, aunt and late uncle whose support and words of encouragement have always directed me towards a successful and fulfilling life.

Acknowledgments

I take this opportunity to express my gratitude for everyone who has helped me during my graduate study at Virginia Tech. First and foremost, I sincerely thank my Ph.D. advisor, Dr. Jerry Park, for his generous support and invaluable guidance in my pursuit of academic excellence. I admire his endeavors to make my graduate study successful and productive. I thank the rest of my doctoral advisory committee: Dr. Tom Hou, Dr. Michael Hsiao, Dr. Daphne Yao, and Dr. Chao Wang for their insightful comments. I thank Dr. Kaigui Bian for his constructive criticism. I thank my colleague and friend, He Li, for his help in writing high quality manuscripts. In my daily work, I have been blessed with a friendly and cheerful group of fellow students in ARIAS, CESCA and Wireless@VT. I would also like to thank my friends in these groups: Behnam, Amr, Seungmo, Sudeep, Jinshan, Gaurang, Pradeep, Pranav, Noah, Aditya, and Munawwar for their contribution to my personal life at Virginia Tech. I enjoyed working with you and have learned a lot from you. I must thank all my friends at Blacksburg, particularly my roommates Dhiraj, Sriram and Navish, and my psuedo-roommates Avik, Prasad, Apoorva, Deepak, and Prashant for making the last five years of my life memorable. I also thank Prabuddha, Santhosh, Krishna, Sarvesh, Bharti, Saurabh, Pratik, Aditi, Gaurav, Ridhhika, and Ravi for their delightful company. Lastly, I thank my father, mother, aunt, brothers, sister-in-laws, father-in-law, mother-in-law, brother-in-law, nephews and nieces for their emotional support. I have my special thanks to my nephews, Pratik and Prafull, to be my special correspondents from my home. Finally, I thank my lovely life-partner, Anamika, for always encouraging me in this journey.

Contents

List of Figures

List of Tables

List of Abbreviations

ACM	Association for Computing Machinery
ASINR	Authentication Signal to Interference and Noise Ratio
ATM	Authentication Tagging using Modulation proposed in [1]
AWGN	Additive White Gaussian Noise
BCNSW	Group signature scheme proposed by Bichsel, Camenisch, Neven, Smart, and Warinschi in [2]
BER	Bit Error Rate
BL-EPID	Enhanced Privacy ID proposed by Brickell and Li in [3]
BN	Barreto-Naehrig
BS	Group signature scheme proposed by Boneh and Shacham in [4]
BTA	Blind Transmitter Authentication
CBAT	Crowd-sourced Blind Authentication of co-channel Transmitters
CDL-EPID	Enhanced Privacy ID proposed by Camenisch, Drijvers and Lehmann in [5]
CEN	Crowd-sourced Enforcement Network
CFO	Carrier Frequency Offset

CP	Cyclic Prefix
CRLB	Cramer-Rao Lower-Bound
DAA	Direct Anonymous Attestation
DDH	Decisional Diffie-Hellman
DFS	Data Fusion Station
DL	Discrete Logarithm
DLIN	Decisional Linear
DSRC	Dedicated Short-Range Communications
DSS	Dynamic Spectrum Sharing
DSSS	Direct Sequence Spread Spectrum
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EFO	Embedded Frequency Offset
EPID	Enhanced Privacy ID
FCC	Federal Communications Commission
FEAT	Frequency offset Embedding for Authenticating Transmitters
FFT	Fast Fourier Transform
FREE	FREquency offset Embedding for CBAT
GS	Group Signature
GSPR	Group Signatures with Probabilistic Revocation
IEC	International Electrotechnical Commission

IEEE	Institute of Electrical and Electronics Engineers
IFFT	Inverse Fast Fourier Transform
IoT	Internet-of-Things
ISI	Inter Symbol Interference
ISO	International Organization of Standardization
LASER	Lightweight Anonymous Attestation with Efficient Revocation
MLSD	Maximum Likelihood Sequence Detection
MRC	Maximal Ratio Combining
MSE	Mean Square Error
MSINR	Message Signal to Interference and Noise Ratio
MSNR	Message Signal to Noise Ratio
NRZ	Non-Return-to-Zero
OFDM	Orthogonal Frequency Division Multiplexing
OOA	Obstruction of Authentication
P-DS	Precoded Duobinary Signaling
P-DSA	Precoded Duobinary Signaling for Authentication
PBC	Pairing-Based Cryptography
PC	Personal Computer
PHY	Physical
PPA	Privacy-Preserving Authentication
PPT	Probabilistic Polynomial Time

PS	Pseudonym-based Signature
PU	Primary User
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
REG	Regulatory Entity
RF	Radio Frequency
RMSE	Root Mean Square Error
RSA	Rivest-Shamir-Adleman
SDH	Strong Bilinear Diffie-Hellman
SDH	Strong Diffie-Hellman
SDR	Software-Defined Radio
SINR	Signal to Interference and Noise Ratio
SNR	Signal to Noise Ratio
SSD	Symbol-by-Symbol Detection
SU	Secondary User
TCG	Trusted Computing Group
TPM	Trusted Platform Module
TSS	Trusted Software Stack
USRP	Universal Software Radio Peripheral
VANET	Vehicular Network
VLR	Verifier-Local Revocation

Chapter 1

Introduction

The exploding demand for radio frequency (RF) spectrum to support wireless applications has motivated spectrum regulatory agencies in industrialized countries to seriously consider and pursue initiatives to realize *dynamic spectrum sharing* (DSS). It is widely believed that a transition from the legacy "command-and-control" spectrum regulatory model—where spectrum is parceled and allocated to specific stakeholders and applications—to a more flexible model of DSS is necessary to achieve more efficient spectrum usage. DSS significantly increases spectrum utilization efficiency, and makes more spectrum available to a greater number of wireless services and users. In the DSS paradigm, the primary users (PUs), who are license holders or incumbents, share the spectrum with the secondary users (SUs) that opportunistically access fallow spectrum not used by the PUs. To harmoniously coexist with PUs as well as other SUs, the SUs need to employ software-defined radios (SDRs) that identify fallow spectrum through spectrum sensing and/or by directions given by spectrum geolocation databases [6]. The SUs also follow a set of prescribed rules or regulations to protect the PUs from interference. Unlike a legacy radio, which is hardware or firmware-based, a SDR enables a SU to readily re-configure its transmission parameters through changes in the software code, allowing for greater flexibility. However, this "programmability" of SDRs also significantly increases the possibility of malicious or "rogue" transmitters that pose a

great threat to other radios. We define a rogue radio as a non-compliant transmitter that violates regulator-prescribed spectrum access rules and regulations.

The problem of rogue transmitters is an especially critical issue in the U.S., where spectrum sharing between federal government, including the military, systems and commercial systems will become a reality in the near future. For example, per its Report and Order (GN Docket 12-354 [7]) published in 2015, the U.S. Federal Communications Commission (FCC) has opened up the 3.5 GHz band to secondary-user access, and has mandated the deployment of technologies to realize spectrum sharing between military radar systems and commercial small-cell networks that will coexist in that band [8]. The harmful interference due to rogue radios poses a serious threat to the federal incumbent users, and is a major security problem that is being actively studied by government and industry stakeholders [9]. Hence, in such spectrum sharing scenarios, rogue radios that effectively hijack spectrum resources or disturb peaceful coexistence need to be thwarted.

One viable approach for deterring rogue transmissions is to enable a regulatory enforcement entity (e.g., FCC's Enforcement Bureau) to uniquely identify transmitters by authenticating their waveforms. This *ex post enforcement* approach would enable the enforcement entity to identify an interference source and collect evidence of interference without burdensome complexity. While cryptographic mechanisms at the higher layers have been widely used to authenticate transmitters, the ability to authenticate and/or uniquely identify transmitters at the PHY-layer is especially useful in heterogeneous coexistence environments, where incompatible systems (i.e., systems with different protocol stacks) may not be able to decode each others' higher-layer signaling—e.g., IEEE 802.22 and 802.11af systems coexisting in TV white space. Note that the objective of transmitter authentication in spectrum sharing environment is to uniquely identify the transmitter that has transmitted a given waveform by authenticating the waveform itself, which is different from authenticating the message carried by the waveform. The latter is handled at the application layer.

For transmitter authentication to be a viable approach for spectrum access enforcement, all

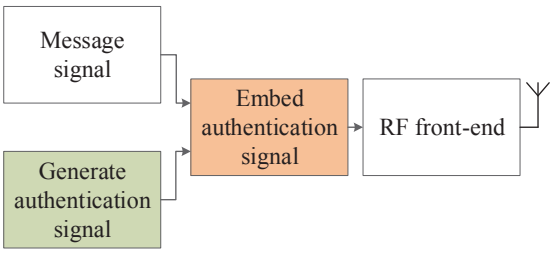


Figure 1.1: Model of a transmitter in spectrum rule enforcement.

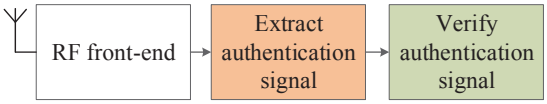


Figure 1.2: Model of an enforcement entity in spectrum rule enforcement.

transmitters should be mandated to employ a authentication-signal-embedding mechanism for embedding an authentication signal into the message signal (which contains the data that the transmitter wants to send). The authentication signal, at a minimum, contains information that enables the enforcement entity to determine the regulator-assigned identity of the transmitter (possibly a certificate of compliance, i.e., FCC *Declaration of Conformity*) as well as the regulator-imposed spectrum access constraints, in terms of frequency, spatial, and temporal domains [1, 9]. Also, tamper resistance mechanism, such as a Trusted Platform Module (TPM), should be employed to prevent hackers from circumventing this the authentication-signal-embedding mechanism [10, 11, 12]. In this dissertation, we assume that an authentication-signal-embedding mechanism as well as a tamper resistance mechanism are incorporated into every radio platform used by a SU. Note that the mandatory use of these mechanisms is consistent with the requirements stipulated in the FCC's Report and Order [7] for realizing federal-commercial spectrum sharing in the 3.5 GHz band.

Figures 1.1 and 1.2 broadly illustrate the operations performed at the transmitter and the enforcement entity, respectively, to enable transmitter authentication in DSS. As shown in Figure 1.1, the transmitter needs to perform two operations, i.e., generate the authentication signal, and embed the authentication signal into the message signal. As shown in Figure 1.2, the enforcement entity needs to perform two operations, i.e., extract the authentication signal from the received signal, and verify the authentication signal. Hence, there are two primary technical problems in devising a transmitter authentication scheme: (1) how to generate

and verify the authentication signal such that the required security criteria are met; and (2) how to embed and extract the authentication signal without negatively impacting the performance of the transmitters and the receivers in DSS. In the following sections, we provide elaborate discussions on these two problems.

1.1 Generate and Verify the Authentication Signal

The notion of authentication is to enable a signer (i.e., a transmitter) to prove its identity to a verifier (i.e., an enforcement entity) and/or to show that she is the origin of the transmitted data. This security attribute is essential to most of today's applications that rely on digital communications over insecure networks. However, in DSS, if the authentication signal contains the unencrypted identity of the transmitter, and is transmitted over-the-air at the PHY-layer, any RF receiver with the knowledge of the authentication embedding and extracting processes can demodulate the raw bits of the authentication signal. This means that the authentication signal can be exploited by eavesdroppers to extract the identity of the transmitters, and monitor or track their transmission behavior, e.g., areas of operation, times of operation, etc. Hence, in DSS, authentication is not sufficient, and in addition to authentication, the transmitter's privacy need to be protected—the combination of these two attributes is often referred to as privacy-preserving authentication (PPA). A wide variety of other applications require PPA including vehicular communication applications [13, 14], and remote attestation of computing platforms [15, 16].

Note that the privacy of a transmitter can also be circumvented by the leakage of side-channel information, e.g., the source and destination addresses included in the frames of the message signal. Hence, novel PHY-layer techniques need to be employed to complement the conventional PPA for a comprehensive approach for preserving privacy of a wireless transmitter in DSS [17]. The discussions of these techniques are out of scope of this dissertation.

For deployment in large networks, PPA protocols need to rely on public-key cryptography. In

public-key cryptosystem-based PPA protocols, there are three entities that interact with each other: *platform*, *verifier*, and *issuer*. For the DSS scenario, the platform is a transmitter and a signer that generates a signature and a corresponding authentication signal. The verifier is an enforcement entity that verifies the received authentication signal and the corresponding signature. The issuer plays an important role. During the initialization process, the issuer generates network parameters, and credentials, certificates (e.g., public-key certificates) or the private/secret signing keys of the platforms. The issuer also revokes compromised or insecure platforms by updating and publishing a revocation list.

The PPA schemes in the existing literature can be classified into two approaches: (1) *verifier-anonymous authentication*; and (2) *full-anonymous attestation*. Specifically in DSS, the platform may employ either of these two approaches based on the regulator-prescribed policy. In the following sub-sections, we discuss these two approaches.

1.1.1 Verifier-Anonymous Authentication

Verifier-anonymous authentication schemes are needed in applications where the verifiers should not learn the actual identity of the platform, and are willing to accept an authentication artifact (i.e., signature) that is verifiably linked to an anonymous platform, knowing that the platform's identity can be revealed by a trusted third party, i.e., the issuer, if disputes need to be resolved. These schemes can be further categorized into two sub-approaches: *pseudonym-based signatures* (PSs) [18, 19, 20] and *group signatures* (GSs) [4].

In PSs, legacy public-key cryptosystems (e.g., RSA) are used. The issuer provides the platform with a list of pseudonyms and the corresponding private keys, public keys, and public-key certificates. The platform creates a signature based on its pseudonym, and replaces its pseudonym with a new one periodically to preserve anonymity. Although the PS approach is straightforward, it has a number of drawbacks. Because each pseudonym needs to be used with its unique set of private and public keys and a certificate, key management and distribution become a very onerous burden in large networks [14].

GSs do not require public-key certificates, and hence do not need a certificate distribution framework. In GS, each platform is a member of a group, and it is provided with a private key tuple by the issuer. Using this tuple, the platform generates signatures without revealing its true identity to the verifier. In the case of a conflict, the signature can be "opened" by the issuer, and the identity of the platform is revealed. The most practical GS schemes support verifier-local revocation (VLR) [2, 4, 21]. To perform VLR, the issuer generates a revocation token for each platform (which is a portion of the private key tuple), publishes it in a revocation list, and distributes the revocation list to the verifiers. To check the revocation status of the private key used to generate the received signature, the verifier performs the revocation check procedure. This procedure involves going through the revocation list, and checking whether any of the revocation tokens contained therein can be mapped to the received signature. This means that the computation time for the revocation check procedure increases linearly with the number of revoked private keys. Hence, the VLR GS schemes in the prior art are not scalable for a large number network (e.g., DSS) with the possibility of a large number of revoked private keys.

As part of this dissertation, we propose a novel VLR GS scheme called *Group Signatures with Probabilistic Revocation* (GSPR) in Chapter 4. As its name implies, the most striking attribute of GSPR is that it supports *probabilistic revocation*. That is, GSPR's revocation check procedure does not produce deterministic results, but instead produces probabilistic results, which may include false positive (i.e., false alarm) results but *no* false negative results. Here, a false negative result refers to an instance in which the revocation check algorithm fails to detect that the revocation token associated with the received signature is included in the revocation list. GSPR includes a procedure that can be used to iteratively decrease the probability of false alarms. The use of probabilistic revocation (instead of deterministic revocation) enables GSPR to elegantly address the primary performance bottleneck of GSs—i.e., enable very efficient revocation checking with only a modest increase in the signature size. In fact, GSPR's revocation check time does not grow linearly with the number of revoked keys.

1.1.2 Full-Anonymous Attestation

The PPA protocols which ensure that neither the verifier nor the issuer can reveal the identity of the platform are categorized as full-anonymous attestation. *Direct Anonymous Attestation* (DAA) is a cryptographic protocol that enables full-anonymous attestation of a computing platform [15, 22, 23]. DAA preserves the privacy of the platform's user by decoupling the information about the platform's configuration and the identity of the platform's user. In DAA, a platform consists of a host and a *trusted platform module* (TPM). The TPM is a secure and dedicated cryptoprocessor which is designed to secure the platform by integrating its cryptographic keys into its hardware [24]. With the help of the host, the TPM generates an anonymous signature on the message corresponding to the current configuration of the platform. The host utilizes the credentials obtained from the issuer to assist the TPM in the generation of signatures by performing most of the computationally expensive operations. The verifier verifies the validity of a signature. As part of the verification process, the verifier checks the revocation status of the platform from which the signature was received.

The Trusted Computing Group (TCG) standardized the RSA-based DAA by including it in its TPM specification version 1.2 [15, 25]. TCG has also included the elliptic curve cryptography (ECC)-based DAA in the most recent version, TPM specification version 2.0 [3, 24, 26]. This TPM specification has also been standardized by the International Organization of Standardization (ISO) and the International Electrotechnical Commission (IEC) [27]. Although the computing industry and academia have made noteworthy strides in improving the security and efficacy of the DAA schemes in recent years, all of these schemes still share a common critical drawback that hinder their widespread adoption. All of the existing DAA schemes suffer from significant computational complexity and communication overhead that *increase proportionally to the size of the revocation list*. This drawback renders them to be impractical when the size of the revocation list grows beyond a relatively modest size.

To support the revocation check procedure, the existing DAA schemes employ a signature-based revocation list [3, 5]. In these schemes, for each tuple in the revocation list, the

platform needs to generate a proof-of-knowledge to prove that its secret key has not been revoked, and include the proof-of-knowledge in each signature as a component of the signature. Hence, three things increase linearly with the number of revoked platforms indicated in the revocation list [28]: (1) the computational overhead of the platform in generating a signature; (2) the computational overhead of the verifier in verifying the validity of the signature; and (3) the length of the signature. This consequence of this attribute poses a significant technical challenge in terms of the implementation and deployment of DAA in real-world applications—i.e., the computation complexity (and to a lesser extent, the communication overhead) becomes unacceptably high for most applications when the length of the revocation list goes beyond a modest number (e.g., a few hundred revoked platforms). Unfortunately, the only solution that has been proposed in the existing literature [16, 28] is a somewhat crude solution for addressing this problem. This approach requires the issuer to "reset" the group that it is managing (i.e., replace all credentials and keys of the platforms with new ones) when the number of tuples in the revocation list exceeds a pre-determined threshold value. This reset procedure may not be acceptable for DAA in large networks or for applications that have stringent latency requirements.

As part of this dissertation, we propose a novel DAA scheme called *Lightweight Anonymous attestation Scheme with Efficient Revocation* (LASER) in Chapter 5. LASER addresses the problem of revocation scalability in an elegant manner. In LASER, the computational complexity and communication overhead of the signature generation and verification procedures are multiple orders of magnitude lower than the prior art. LASER achieves this significant performance improvement by shifting most of the computational complexity and communication overhead (due to the revocation check procedure) from the DAA's *online* procedure (i.e., signature generation and verification) to its *offline* procedure (i.e., acquisition of keys/credentials from the issuer). We assert that this strategy significantly improves the practicality of DAA in real-world applications, because the critical performance bottlenecks of those applications are determined by the performance of the online procedure.

Unlike existing DAA schemes, in LASER, the platform does not need to include any proof

of non-revocation of its secret key in the signatures sent to the verifier. This unique feature of LASER brings about a number of important practical advantages. First, during the signature generation procedure, the platform is not burdened with any computations related to the revocation check procedure, resulting in a significant reduction in the computational complexity of signature generation. Second, the signature length is constant, and does not grow proportionally with the length of the revocation list. Third, LASER enables the verifier to employ a computationally efficient procedure to check the revocation status of the platform that has issued a given signature. These advantages are especially important when a DAA scheme needs to be deployed in a network with a large number of nodes. Unlike legacy DAA schemes, LASER is scalable, and can be deployed in DSS networks, which are expected to have long revocation lists.

1.2 Embed and Extract the Authentication Signal

The PHY-layer authentication-signal-embedding schemes in the existing literature [1, 29, 30, 31, 32] can be classified into two categories: (1) *intended receiver-based authentication* (IRA); and (2) *blind transmitter authentication* (BTA). In this dissertation, in addition to these two categories, we discuss a new category of PHY-layer authentication schemes called *crowd-sourced blind authentication of co-channel transmitters* (CBAT). In the following subsections, we discuss these three categories.

1.2.1 Intended Receiver-Based Authentication

In most of the PHY-layer authentication schemes in the existing literature [1, 29, 30], the message signal is modified to embed the authentication signal in such a way that the enforcement entity needs to decode/demodulate the message signal to extract the authentication signal. Hence, the enforcement entity can extract the authentication signal from the received signal if it is also the "intended receiver" of the message signal. Here, the intended receiver

denotes the receiver which coordinates with the transmitter to obtain the information about the transmission parameters and protocols so that it can demodulate and decode the message signal. These schemes can be categorized as IRA.

In most of the existing schemes in this category, the authentication signal is added to the message signal such that the authentication signal appears as noise to the message signal and vice versa—we refer to this approach as the "*blind signal superposition*" method [33]. In this approach, the authentication signal is fully present when the message signal is decoded, thus resulting in decreased signal to interference and noise ratio (SINR) for the message signal, assuming that the transmission power has not been increased to embed the authentication signal. Hence, there is a fundamental tradeoff between the message signal's SINR and the authentication signal's SINR—it is impossible to improve the former without worsening the latter and vice versa. This means that the degradation in the message signal's SINR is significant when the authentication signal's SINR is increased to a level sufficient for authenticating the received signal at the receiver [34].

To overcome this trade-off, we propose a novel IRA scheme, called *Precoded Duobinary Signaling for Authentication* (P-DSA) in Chapter 6. Our approach exploits the inherent redundancy in pulse shaping to embed the authentication signal into the message signal without being constrained by the aforementioned tradeoff. Specifically, our approach uses the redundancy in *duobinary signaling* which is a waveform shaping technique that has been traditionally used to increase bandwidth efficiency [35, 36].

1.2.2 Blind Transmitter Authentication

The IRA schemes enable transmitter authentication in heterogeneous coexistence environment. However, in spectrum sharing environment, the enforcement entity that is attempting to identify the non-compliant or rogue transmitter is *not* the intended receiver. Hence, we refer to such a receiver as a "blind receiver". As the name implies, the blind receiver has little, if any, knowledge about the communication parameters needed to demodulate and

decode the detected signal. Hence, the blind receiver would need to carry out transmitter authentication at the PHY-layer, where the least amount of knowledge of the communication parameters is needed to authenticate the transmitter. We coin the term *Blind Transmitter Authentication* (BTA) to refer to the problem of authenticating a transmitter by extracting its unique, identifiable information from the received signal with *little* or *no* knowledge of the transmission parameters.

We want to emphasize that there are a few important differences between a BTA scheme [31, 32] and the conventional IRA schemes. In the latter schemes, it is assumed that the receiver (that is authenticating the signal) has complete knowledge of the transmission parameters, whereas in the former scheme, the receiver is "blind". Moreover, most, if not all, of the IRA schemes are designed to work when the received signal's SINR is sufficiently high—e.g., high enough to demodulate and decode the message signal correctly. Because a blind receiver is not the "intended" receiver, it may need to carry out BTA at a location where the SINR is very low with significant multipath fading. Conventional PHY-layer authentication schemes would perform very poorly under such conditions. An *ideal* BTA scheme satisfies two requirements: (1) it enables a receiver to "blindly" extract the authentication information from the signal with little or no knowledge of the transmission parameters; and (2) authentication can be performed under very harsh conditions (i.e., low SINR and significant multipath fading).

As part of this dissertation, we propose a BTA scheme called *Frequency offset Embedding for Authenticating Transmitters* (FEAT) in Chapter 7. To the best of our knowledge, FEAT is the first scheme that satisfies the two requirements of an ideal BTA scheme. FEAT modifies the frequency offset of each frame of the message signal to embed the authentication signal into the message signal. This is achieved in such a way that the authentication signal does not interfere with the decoding process of the message signal. Also, the authentication signal can be estimated at the blind receiver with only limited knowledge about the transmission parameters by estimating the frequency offset of each frame.

1.2.3 Crowd-Sourced Blind Authentication of Co-channel Transmitters

To successfully carry out transmitter authentication in DSS, an enforcement entity faces three real-world challenges. Firstly, the enforcement entity is considered a blind receiver which denotes a receiver that has little, if any, knowledge of the transmission parameters needed to demodulate and decode the received waveforms. Secondly, the enforcement entity needs to cope with the possibility of multiple simultaneous transmissions. There may be simultaneous transmissions from multiple transmitters operating in the same frequency that are located within the reception range of an enforcement entity. In such a situation, the signals received at the blind receiver may contain multiple unique authentication signals, which need to extracted and separated. Thirdly, the enforcement entity may need to cope with the reception of very poor-quality signals that have a very low SINR. Note that the vast majority of the existing PHY-layer authentication schemes including P-DSA are designed to work only when the SINR of the received signal is sufficiently high, i.e., high enough to demodulate and decode the message signal correctly [1, 29]. The first and third challenges have been addressed using FEAT and by the prior art [31, 32]. However, to the best of our knowledge, none of the schemes reported in the current literature adequately addresses the second challenge.

To carry out BTA in real-world deployment scenarios, we need a network of enforcement nodes, collaborating with each other to perform transmitter authentication. Unfortunately, deploying and maintaining a network of *dedicated* enforcement nodes for this purpose is prohibitively expensive [37]. There is a more economically viable alternative. This approach involves the use of a limited number of dedicated enforcement nodes, and the employment of a much greater number of SUs' radios that act as *non-dedicated* enforcement nodes to greatly enhance the enforcement capability of the dedicated nodes. We refer to a network of dedicated and non-dedicated enforcement nodes as a *crowd-sourced enforcement network* (CEN). In the CEN, the SU radios use their spare resources to act as non-dedicated en-

forcement nodes in exchange for a well-defined payoff, e.g., monetary remuneration. The incentives that can act as payoffs are discussed in [38]. We do not discuss the issue of payoff any further as it is outside the scope of this dissertation.

In this dissertation, we investigate the idea of *Crowd-sourced Blind Authentication of co-channel Transmitters* (*CBAT*). Specifically, we consider CBAT in a scenario where a CEN consists of a data fusion station (DFS), and a number of dedicated and non-dedicated enforcement nodes. Note that all nodes in the CEN can be considered as blind receivers. In the first phase of CBAT, each blind receiver extracts the authentication information from its received signals, and then sends its results to the DFS. In the second phase, the DFS performs data fusion to integrate the results collected from the blind receivers to authenticate one or more transmitters whose signals have been recorded.

In this dissertation, we propose a concrete instantiation of CBAT called *FREquency offset Embedding for CBAT* (*FREE*). In FREE, the transmitter's authentication information is embedded into the waveform as a series of controlled frequency offsets. CBAT is performed in a distributed manner by a CEN. The DFS performs data fusion in such a way as to maximize the probability of successful CBAT. According to our findings, FREE is very effective in addressing all of the aforementioned challenges. To the best of our knowledge, none of the existing schemes can make the same claim.

1.3 Contributions

In this dissertation, we analyze two primary technical problems in devising a transmitter authentication scheme for DSS: (1) how to generate and verify the authentication signal; and (2) how to embed and extract the authentication signal. For solving the first problem, we propose GSPR in Chapter 4 which is a verifier-anonymous authentication scheme. Considering a more strict privacy criteria of full-anonymous attestation, we propose LASER in Chapter 5. For solving the second problem, in Chapter 6, we propose P-DSA which,

similar to the prior art, assumes that the enforcement entity is also an intended receiver. In Chapter 7, the proposed scheme, FEAT, gets rid of this assumption, and considers a more practical network scenario where the enforcement entity is considered a blind receiver. Lastly, we propose FREE in Chapter 8 by extending the concepts of FEAT for crowd-sourced enforcement network which achieves significant advantage over the prior art in terms of the robust detection of the transmitter's authentication signal.

The main contributions of this dissertation are summarized below.

1. We propose a novel group signature (GS) scheme called Group Signatures with Probabilistic Revocation (GSPR) which significantly reduces the computational complexity of the revocation check procedure compared to the prior art. We also propose the novel concept of probabilistic revocation which makes an advantageous tradeoff between computational complexity and communication overhead. This tradeoff enables GSPR to have significantly better scalability in terms of revocation compared to the prior art.

2. We propose a novel direct anonymous attestation (DAA) scheme called Lightweight Anonymous attestation Scheme with Efficient Revocation (LASER), which significantly reduces the computational and communication complexity of the signature generation and verification procedures compared to the prior art.

3. We propose a intended receiver-based authentication (IRA) scheme called Precoded Duobinary Signaling for Authentication (P-DSA), which does not suffer from the drawbacks of the blind signal superposition.

4. We define the challenges in the blind transmitter authentication (BTA), and propose a BTA scheme called Frequency offset Embedding for Authenticating Transmitters (FEAT). We demonstrate that FEAT is the first scheme that satisfies all of the required criteria of an ideal BTA scheme.

5. We define the challenges in crowd-sourced blind authentication of co-channel transmitters (CBAT), and propose a CBAT scheme called FREquency offset Embedding for CBAT (FREE). According to our results, FREE outperforms the existing PHY-layer authentication approaches in all of the performance criteria that were considered.

The work presented in this dissertation has resulted into the following papers.

- **Journal**

  1. **V. Kumar**, J.-M. Park, and K. Bian, "Transmitter authentication using hierarchical modulation in dynamic spectrum sharing," under review.
  2. **V. Kumar**, J.-M. Park, and K. Bian, "PHY-layer authentication using duobinary signaling for spectrum enforcement," *IEEE Transactions on Information Forensics and Security*, vol.11, no.5, pp.1027-1038, May 2016.
  3. J.-M. Park, J. H. Reed, A. A. Beex, T. C. Clancy, **V. Kumar**, and B. Bahrak, "Security and enforcement in spectrum sharing," *Proceeding of the IEEE*, vol.102, no.3, pp.270-281, March 2014 (invited).

- **Conference**

  1. **V. Kumar**, H. Li, J.-M. Park, and K. Bian, "Enforcement in spectrum sharing: Authentication of waveforms from simultaneous co-channel transmissions," under review.
  2. **V. Kumar**, H. Li, P. Asokan, N. Luther, and J.-M. Park, "LASER: Lightweight anonymous attestation scheme with efficient revocation," under review.
  3. **V. Kumar**, H. Li, J.-M. Park, K. Bian, and Y. Yang "Group signatures with probabilistic revocation: A computationally-scalable approach for providing privacy-preserving authentication," in *ACM Conference on Computer and Communications Security (CCS)*, pp.1334-1345, Oct. 2015.

4. **V. Kumar**, J.-M. Park, and K. Bian, "Blind transmitter authentication for spectrum security and enforcement," in *ACM Conference on Computer and Communications Security (CCS)*, pp.787-798, Nov. 2014.

5. **V. Kumar**, J.-M. Park, T. C. Clancy, and K. Bian, "PHY-layer authentication using hierarchical modulation and duobinary signaling," in *International Conference on Computing, Networking and Communications (ICNC)*, pp.782-786, Feb. 2014.

6. **V. Kumar**, J.-M. Park, T. C. Clancy, and K. Bian, "PHY-layer authentication by introducing controlled inter symbol interference," in *IEEE Conference on Communications and Network Security (CNS)*, pp.10-18, Oct. 2013.

Chapter 2

Related Work

In this chapter, we discuss the existing literature in two sections corresponding to the schemes for solving the two primary technical problems in devising a transmitter authentication scheme, i.e., how to generate and verify the authentication signal, and how to embed and extract the authentication signal.

2.1 Generate and Verify the Authentication Signal

The design of the authentication signals proposed in the existing PHY-layer authentication schemes [1, 33, 31] for spectrum enforcement, pose a potentially serious threat to the privacy of the the signer/platform. Here, for preserving the privacy of the platform using verifier-anonymous authentication, we discuss the approach of group signatures (GSs) [2, 4, 14, 39, 40, 41, 42, 43, 44]. For the full-anonymous authentication, we discuss the approach of direct anonymous attestation (DAA) [3, 5, 15, 16, 45, 46].

2.1.1 Group Signatures

The GS based schemes in the recent literature can be divided into two categories based on their revocation check procedures. In the first category of techniques, the revocation check procedure takes place at the platforms [42, 43, 44]. The scheme proposed in [44] achieves constant signing and verification time at the cost of the public key of $O(\sqrt{n})$-size, where n is the total number of platforms in the network. In [42], although the signing and verification have constant time along with constant-size group public key, the computational cost at the issuer grows with $O(n^2)$ which means that the issuer becomes the bottleneck. The scheme proposed in [43] achieves constant cost for signing and verification without significantly increasing the size of the public key. However, the length of each signature in [43] is significantly large, e.g., around 20 times that in [4]. While the schemes proposed in [42, 44] are secure in the random oracle model, the scheme proposed in [43] is constructed in the standard model.

In the second category of schemes, the revocation check procedure takes place at the verifier through verifier-local revocation (VLR) [2, 4, 21]. In these schemes, it is the responsibility of the verifier to check whether a platform has been revoked or not, by using the revocation list which contains the revocation tokens corresponding to the revoked private keys. However, the computational cost of revocation check procedure in these schemes increases linearly with the number of revoked private keys. These schemes are secure in the random oracle model.

2.1.2 Direct Anonymous Attestation

For the full-anonymous authentication, Brickell, Camenisch and Chen [15] proposed the concept of DAA, and the first instantiation of DAA. This work was followed by a number of enhancements in the DAA scheme. The most notable RSA based DAA scheme is called the Enhanced Privacy ID (EPID) [16]. The ECC based DAA schemes [3, 45, 47, 48, 49, 50] are

shown to be significantly more efficient in terms of the computational and communication overheads as compared to the RSA based DAA schemes. Some DAA schemes with additional features have also been proposed in the existing literature, e.g., DAA with attributes [51].

In terms of security of the DAA schemes, the issue of subgroup membership was highlighted in [52, 53], and the notion of user-controlled linkability was discussed in [46]. In the existing literature, most of the ECC based DAA schemes are proved secure in the model presented in [45]. However, Camenisch et. al. [5, 54] have identified specific shortcomings in the security models and security properties utilized in the DAA scheme. They have proposed a new security model and an instantiation of a secure DAA scheme.

Brickell and Li [16], and Chen and Li in [28] provide a comprehensive picture of various cases where different types of revocation mechanisms are required. The only solution that is provided to keep the DAA scalable is to utilize *rekey*-based revocation, i.e., to reset the whole group when the revocation list becomes long. However, the group resetting scheme is trivial, and is less likely to work in a large network with a large number of platforms. A few computationally efficient DAA schemes utilize the idea of revocation based on *reputation* of a platform [55, 56]. However, the reputation based revocation allows a time period for the platform in which it can generate *some* number of valid signatures before it is actually revoked. Notably, the damage caused by these signatures (which should have ideally been invalid) is unaccounted, and may vary based on the applications.

Arguably, it is desirable that the revocation check procedure takes place at the verifier through VLR in DAA as in the case of group signature schemes [4]. Unlike group signatures, DAA signatures are anonymous not only to the verifiers, but also to the key/credential issuer. Due to this feature, it is more difficult to design a DAA scheme than to design a group signature scheme with VLR which achieves efficient revocation.

In terms of the implementation of DAA, there are only a few research works in the existing literature. In [57], the authors illustrate the functions and algorithms needed to implement DAA using TPM. In [58], the authors discuss the performance of multiple DAA schemes on

various elliptic curves with different parameters. Chen et. al. [59] provide a comprehensive computational overhead analysis of the existing DAA schemes. However, in these papers, the functionality of the TPM is only simulated which means that these papers do not provide significant insight into the performance of a real-implementation of DAA using TPM.

2.2 Embed and Extract the Authentication Signal

In essence, the techniques for PHY-layer embedding and extraction of the authentication signal is closely related to radio frequency (RF) fingerprinting [60, 61, 62], electromagnetic signature identification [63, 64, 65], PHY-layer watermarking [33, 66, 67, 68], transmitter identification [69, 70], and transmitter authentication [1, 29, 69, 70, 71, 72]. These schemes can be broadly divided into two categories: intrinsic and extrinsic approaches.

2.2.1 Intrinsic Approach

The schemes in this category utilize the transmitter-unique "*intrinsic*" characteristics of the waveform as unique signatures to authenticate/identify transmitters. They include RF fingerprinting, and electromagnetic signature identification [61, 62, 63, 64]. Although these intrinsic approaches have been shown to work in controlled lab environments, their sensitivity to factors—such as temperature changes, channel conditions, and interference—limit their efficacy in real-world scenarios. Moreover, it has been shown in [73] that the identification of a platform based on transmission imperfections exhibited by its radio transmitter, is prone to impersonation attacks. In this way, the intrinsic approaches require the blind receiver to have only a little knowledge about the transmission parameters to authenticate the transmitter, but they are limited by their low robustness against noise and security attacks.

2.2.2 Extrinsic Approach

The schemes in the second category enable a transmitter to "*extrinsically*" embed an authentication signal (e.g., digital signature) in the message signal and enable a receiver to extract it. In the approach, called blind signal superposition [33], the authentication signal is embedded in the message signal in such a way that the authentication signal acts as noise to the message signal and vice versa [1, 29, 30]. As mentioned previously, this method is constrained by the unavoidable tradeoff between the message signal's SINR and the authentication signal's SINR. To limit the detrimental effects of the authentication signal on the message signal, the principle of hierarchical modulation is often applied—i.e., the authentication signal (low priority signal) is carried on the low-power, high-resolution constellation while the message signal (high priority signal) is embodied by the high-power, low-resolution constellation. In such an approach, the message signal is decoded in the presence of the authentication signal, thus resulting in decreased SINR for the message signal, assuming average transmission power has not been increased to embed the authentication signal. This means that the degradation in the message signal's SINR is significant when the authentication signal's SINR is increased to a level sufficient for authenticating the embedded signal at the receiver [34]. Hence, there is a fundamental tradeoff between the message signal's SINR and the authentication signal's SINR.

A number of previous studies have attempted to address the BTA problem in different ways, and among them the schemes proposed in [31, 32, 33, 72] are most noteworthy. The authors in [33, 72] propose authentication schemes in which the authentication signal is embedded into the message signal extrinsically to modify an intrinsic characteristic (channel characteristics or carrier frequency offset) of the message signal. This enables the blind receiver to decode the authentication signal with high robustness with only a little knowledge about the transmission parameters. In [33], the message signal at the transmitter is processed with a synthesized channel-like filter that is generated using the authentication signal. However, since this approach requires estimation of the channel response at the receiver, it may not be

a viable approach when the coherence time is short. The PHY-layer authentication scheme in [72] embeds the authentication signal as a frequency shift in the pilots of the message signal. This scheme affects the performance of the channel estimation process at the receiver. The scheme proposed in [32] embeds the authentication by varying the size of the cyclic prefix (CP) in each symbol of the transmitted signal using orthogonal frequency division multiplexing (OFDM). However, this scheme hampers the ability of the transmitted signal to cope with the inter-symbol interference. The authors in [31] propose an authentication scheme in which the authentication signal is embedded into the message signal by inducing a cyclo-stationary signature through repetition of the same symbols over multiple sub-carriers. This scheme achieves authentication at the cost of lowering the message throughput.

We note that none of the schemes in the prior art can be utilized to address the problem of CBAT. However, CBAT is closely related to the research domain of crowd-sourced spectrum sensing [37, 74, 75, 76]. In most of the existing literature on crowd-sourced spectrum sensing, the crowd-sourced receivers are utilized for detecting the presence of a rogue transmission in a channel by conventional energy detection. In these schemes, no authentication/identification signal is embedded in the transmitted signal. Hence, these schemes are not suitable for transmitter authentication in enforcement applications.

Chapter 3

Technical Background

In the following sections, we present the technical background required for the transmitter authentication schemes proposed in this dissertation.

3.1 Bilinear Mapping

A pair of multiplicative cyclic groups of prime order p, $\mathbb{G}_1$ and $\mathbb{G}_2$, is called a bilinear group pair, if there exists a group $\mathbb{G}_T$, and a bilinear mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ with the following properties:

1. Computable: $e(u, v)$ is efficiently computable for all $u \in \mathbb{G}_1$, and $v \in \mathbb{G}_2$.
2. Bilinear: $e(u^\alpha, v^\beta) = e(u, v)^{\alpha \cdot \beta}$, for all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$, and $\alpha, \beta \xleftarrow{R} \mathbb{Z}_p^*$. Here, $\mathbb{Z}_p^*$ represents the set of integers modulo p, and $\xleftarrow{R}$ represents a random selection.
3. Non-degenerate: $e(g_1, g_2) \neq 1$, where g_1 and g_2 are the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively.

Based on the relationship between the groups $\mathbb{G}_1$ and $\mathbb{G}_2$, there are three types for bilinear mappings as mentioned below and elaborately discussed in [77].

1. Type-1: $\mathbb{G}_1 = \mathbb{G}_2$.
2. Type-2: $\mathbb{G}_1 \neq \mathbb{G}_2$, but there exists an efficiently computable isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$. Certain families of non-supersingular elliptic curves can be used for efficient implementation of bilinear groups, and the isomorphism ψ can be implemented by a trace map [78].
3. Type-3: $\mathbb{G}_1 \neq \mathbb{G}_2$, and there does not exist any computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$.

We assume that all the elements in the algorithms and protocols are checked for the membership of their specified groups to thwart small-subgroup attacks [52, 53].

3.2 Hash Function

In this dissertation, we assume that $H_z : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ represents a collision resistant hash function that is treated as a random oracle. The hash function H_z can be implemented using conventional hash algorithms, such as SHA-3 [79].

Further, we assume that a group $\mathbb{G}_1$ is constructed using the curve $y^2 = x^3 + a_c x + b_c$, where a_c and b_c are constant parameters. With an input $t \xleftarrow{R} \mathbb{Z}_p^*$, a function $H_g : \mathbb{Z}_p^* \rightarrow \mathbb{G}_1$, can be computed as follows.

1. Set $i = 0$.
2. Compute $x = H_z(t, i)$, and $z = x^3 + a_c x + b_c$.
3. Compute $y = \sqrt{z}$. If y does not exist, set $i = i + 1$, and start back from Step 2.
4. Output (x, y).

3.3 Proof of Knowledge

We utilize proof of knowledge protocols which prove the knowledge of discrete logarithms and the validity of relations among them without revealing any more information about them. These proofs of knowledge utilize the notations and the discussions in [80, 81]. In the random oracle model, a signature scheme can be designed with such proof of knowledge protocol using Fiat-Shamir heuristic [82]. We utilize the following well-known proofs of knowledge.

3.3.1 One Discrete Logarithm

A proof of knowledge of a discrete logarithm, $\alpha \in \mathbb{Z}_p^*$, of an element, $T \in \mathbb{G}_1$, with respect to a base, $u \in \mathbb{G}_1$, is denoted by $PK\{(\alpha) : T = u^\alpha\}$. We represent a signature on a message M obtained in this way as $SPK\{(\alpha) : T = u^\alpha\}(M)$.

3.3.2 Multiple Discrete Logarithms

A proof of knowledge of discrete logarithms, $\alpha_1, \cdots, \alpha_l \in \mathbb{Z}_p^*$, of the representation of an element, $T \in \mathbb{G}_1$, with respect to corresponding bases, $h_1, \cdots, h_l \in \mathbb{G}_1$, is denoted as $PK\{(\alpha_1, \cdots, \alpha_l) : T = h_1^{\alpha_1} \cdot h_2^{\alpha_2} \cdots h_l^{\alpha_l}\}$.

3.3.3 Equality of Discrete Logarithms

The proof of knowledge of the equality of discrete logarithms, represented by $\alpha \in \mathbb{Z}_p^*$, of two elements, $T_1, T_2 \in \mathbb{G}_1$, with respect to two bases, $h_1, h_2 \in \mathbb{G}_1$, respectively, is represented as $PK\{(\alpha) : T_1 = h_1^\alpha, T_2 = h_2^\alpha\}$.

3.3.4 Inequality of Discrete Logarithms

The proof of knowledge of the inequality of two discrete logarithms of two elements, $T_1, T_2 \in \mathbb{G}_1$, with respect to two bases, $h_1, h_2 \in \mathbb{G}_1$, respectively, given that $T_1 = h_1^{\alpha}$, where $\alpha \in \mathbb{Z}_p^*$, is represented as $PK\{(\alpha, \tau) : \nu = \tau \cdot \alpha,\ T_1 = h_1^{\alpha},\ P = h_2^{\nu} \cdot T_2^{-\tau}\}$, where $\tau \xleftarrow{R} \mathbb{Z}_p^*$, and $P \neq 1$.

3.4 Cryptographic Assumptions

The security of the proposed schemes in this dissertation are proved in the random oracle model using the Discrete Logarithm (DL) assumption [83], the Decisional Diffie-Hellman (DDH) assumption [84], the Decisional Linear (DLIN) assumption [40], the q-Strong Diffie-Hellman (q-SDH) assumption in Type-3 bilinear mapping [85], and the Bilinear q-Strong Diffie-Hellman (q-BSDH) assumption in Type-2 bilinear mapping [86, 87]. Here, we provide the definitions of these complexity assumptions.

Assumption 3.1. *($\mathbb{G}_1$-**DL Assumption**): Given $(u, u^a) \in \mathbb{G}_1^2$, where $a \in \mathbb{Z}_p^*$, as input for each probabilistic polynomial time (PPT) algorithm $\mathcal{A}$, the probability that $\mathcal{A}$ outputs a is negligibly small.*

Assumption 3.2. *($\mathbb{G}_1$-**DDH Assumption**): Given $(P, P^a, P^b, P^c) \in \mathbb{G}_1^4$, where $a, b \in \mathbb{Z}_p^*$, as input for each PPT algorithm $\mathcal{A}$, the probability with which $\mathcal{A}$ is able to differentiate whether $c = a \cdot b$, or $c \xleftarrow{R} \mathbb{Z}_p^*$, is negligibly small.*

Assumption 3.3. *($\mathbb{G}_2$-**DLIN Assumption**): Given $(u_0, u_1, h,\ u_0^a,\ u_1^b, Z) \in \mathbb{G}_2^6$, where $a, b \in \mathbb{Z}_p^*$, as input for each PPT algorithm $\mathcal{A}$, the probability with which $\mathcal{A}$ is able to differentiate whether $Z = h^{a+b}$, or $Z \xleftarrow{R}, \mathbb{G}_2$ is negligibly small.*

Assumption 3.4. *(q-**SDH Assumption**): Given a $(q+3)$-tuple $(g_1, g_1^{\gamma}, \cdots, g_1^{\gamma^q}, g_2, g_2^{\gamma})$, where $g_1 \xleftarrow{R} \mathbb{G}_1$, $g_2 \xleftarrow{R} \mathbb{G}_2$, and $\gamma \xleftarrow{R} \mathbb{Z}_p^*$, as input for each PPT algorithm $\mathcal{A}$, the probability that $\mathcal{A}$ outputs a pair $(g_1^{\frac{1}{\gamma+z}}, z)$, where $z \in \mathbb{Z}_p^*$, is negligibly small.*

Assumption 3.5. ***(q-BSDH Assumption)****: Given a* $(q+2)$*-tuple* $(g_1, g_2, g_2^{\gamma}, \cdots, g_2^{\gamma^q})$*, where* $g_1 \xleftarrow{R} \mathbb{G}_1$*,* $g_2 \xleftarrow{R} \mathbb{G}_2$*, and* $\gamma \xleftarrow{R} \mathbb{Z}_p^*$*, as input for each PPT algorithm* $\mathcal{A}$*, the probability that* $\mathcal{A}$ *outputs a pair* $(e(g_1, g_2)^{1/(\gamma+z)}, z)$*, where* $z \in \mathbb{Z}_p^*$*, is negligibly small.*

3.5 Performance Criteria

We present a set of performance criteria which can be used to qualitatively and quantitatively evaluate PHY-layer authentication schemes for embedding and extracting the authentication signal. We will use them to evaluate the proposed schemes, and compare their performance with the prior art.

3.5.1 Overhead

Embedding the authentication signal in the message signal requires applying changes to the message signal itself, and thus incurs some type of PHY-layer overhead when a transmitted signal with authentication signal is compared to a transmitted signal without authentication signal. Examples of such overhead include disadvantageous changes in transmission power, drop in message throughput, increase in bandwidth, and increase in the complexity of the transmitter and intended receivers. Ideally, the PHY-layer authentication should not cause significant overhead related to various aspects.

3.5.2 Compatibility

This criterion dictates that a PHY-layer scheme should embed the authentication signal into the message signal such that it enables an enforcement entity to extract the authentication signal, while at the same time, enables the intended receiver to recover the message signal *without* requiring the intended receiver to change its demodulation or decoding procedure. This is an important criterion in terms of evaluating a scheme's real-world validity, because

a non-compatible scheme would require all receivers—including those that do not need to authenticate the received signals—to modify their demodulation/decoding procedure, which would be prohibitively expensive in some cases.

3.5.3 Message Signal's Error Performance

This criterion determines the message signal's error performance at the intended receiver. Ideally, the intended receiver should be able to extract the message signal from the received signal, and there should not be any degradation in the error performance of the message signal due to embedding of the authentication signal.

3.5.4 Authentication Signal's Error Performance

This criterion determines the authentication signal's error performance at the intended receiver and the enforcement entity. Ideally, the enforcement entity should be able to extract the authentication signal from the received signal even in harsh channel conditions (i.e., very low SNR and significant multipath).

3.5.5 Authentication Rate

The authentication rate is defined as the amount of authentication information (computed in bits) that can be transmitted per second. The authentication signal is embedded by altering the message signal in a certain manner so that the enforcement entity can detect the alteration and use it to extract the authentication information. The rate at which the alteration can be made determines the authentication rate. Usually, the message rate (or message throughput) affects the authentication rate.

3.5.6 Authentication of Concurrent Transmissions

This criteria considers the feasibility of authentication of multiple transmitters which are transmitting concurrently. This means that if multiple transmitters are transmitting on the same spectrum band at the same time, the enforcement entity should be able to uniquely extract the authentication signals corresponding to each of the transmitters from the received signal.

3.5.7 Blind Authentication

The enforcement entity may not the intended recipient of the transmitted signal. Hence, it may not know the transmission parameters, e.g., frame format, preamble samples, modulation scheme, and pilot samples. However, the enforcement entity needs to be able to verify the authentication signal. This criterion takes into account the minimum amount of information needed by the enforcement entity to extract the authentication signal from the received signal.

3.5.8 Security

At the PHY-layer, *integrity* is the only facet of security that needs to be considered. To ensure integrity, the authentication scheme should prevent an adversary from modifying/corrupting the authentication signal of the transmitter without being detected by the enforcement entity.

Chapter 4

GSPR: Group Signatures with Probabilistic Revocation

In this chapter, we propose a novel verifier-anonymous authentication scheme called the *Group Signatures with Probabilistic Revocation* (GSPR), which significantly reduces the computational complexity of the revocation check procedure compared to the prior art. GSPR employs the novel notion of *probabilistic revocation*, which enables the verifier to check the revocation status of the private key of a given signature very efficiently. However, GSPR's revocation check procedure produces probabilistic results, which may include false positive results but *no* false negative results. GSPR includes a procedure that can be used to iteratively decrease the probability of false positives. GSPR makes an advantageous tradeoff between computational complexity and communication overhead, resulting in a GS scheme that offers a number of practical advantages over the prior art. We provide a proof of security for GSPR in the random oracle model using the $\mathbb{G}_2$-DLIN assumption and the q-BSDH assumption discussed in Section 3.4.

The rest of this chapter is organized as follows. We provide the overview of GSPR in Section 4.1, and present the model and security definitions in Section 4.2. We present the details of GSPR in Section 4.3, and analyze its security properties in Section 4.4. We

perform the computational and communication overhead analysis of GSPR in Section 4.5. We discuss GSPR in the context of safety applications for vehicular networks in Section 4.6. We conclude the chapter in Section 4.7.

4.1 Overview of GSPR

In the GSs supporting VLR [2, 4, 21], the issuer includes a revocation token corresponding to each revoked private key in a revocation list, and distributes the revocation list to the verifier. In each VLR based GS scheme, there is an associated implicit tracing algorithm which utilizes the revocation token to link a signature to a revoked private key using which the signature is generated. This implicit algorithm requires several exponentiation and/or bilinear map operations which are computationally expensive. In the revocation check procedure, the verifier performs this implicit tracing algorithm between the received signature, and each revocation token in the revocation list. This means that the computation time for the revocation check procedure of a signature increases linearly with the number of revoked private keys. Hence, the revocation check procedure becomes the major bottleneck in the application of VLR based GSs in real systems with large number of platforms along with possibility of large number revoked private keys.

In this chapter, we propose a VLR based GS, called *Group Signatures with Probabilistic Revocation* (GSPR), in which an alias token is embedded into the group signature generated by a platform in such a way that it can be utilized for the purpose of revocation check procedure. GSPR significantly reduces the computation complexity of the revocation check procedure by adopting two techniques. Firstly, it reduces the computation cost of executing the implicit tracing algorithm by using the alias tokens in generating signatures. Secondly, it enables the verifier to check the revocation status of an alias token in a single step, instead of requiring the verifier to sequentially go through the revocation list and execute the implicit tracing algorithm for each revocation token included in the revocation list.

Specifically, the dramatic improvement in the computational efficiency of the revocation check procedure is made possible by the use of "alias codes". Each alias code is a vector of $+1$s and -1s with desirable cross-correlation properties, and each alias code is mapped to an alias token (which is equivalent to a revocation token in legacy VLR GS schemes) included in each signature. The issuer issues a set of alias tokens corresponding to a private key of the platform, and the platform embeds an alias token in each of its generated signatures. The issuer creates a "revocation code" (which is equivalent to a revocation list) by computing sample-by-sample addition of all of the alias codes mapped to revoked alias tokens. When the private key of a platform is revoked, all its corresponding alias tokens are mapped to the corresponding alias codes. The issuer performs sample-by-sample addition of all the alias codes corresponding to the revoked alias tokens to generate a code which is added to the revocation code. The revocation code, instead of the revocation list, is distributed to the verifier. When the verifier receives a particular signature with a particular alias token, it generates the alias code corresponding to the alias token. The verifier computes the cross correlation of the alias code and the revocation code. If the value of correlation exceeds a particular threshold, the verifier presumes that the alias code (of the signature being verified) is used to generate the revocation code, and in turn concludes that the signature is invalid because it is associated with a revoked alias token. Otherwise, the verifier concludes that the signature is valid. Note that the verifier is able to check whether a particular alias code is included in the revocation code in a *single* cross-correlation operation, and thus avoids the burden of legacy GS schemes in which the verifier needs to iteratively check each revocation token in the revocation list. Because of the probabilistic nature of the revocation check procedure, its result is not guaranteed to be correct with certainty, but only with a certain probability.

The motivation behind the concept of the alias codes and the probabilistic revocation, which is one of the distinguishing attributes of GSPR comes from direct-sequence spread spectrum (DSSS) systems used in communications [88]. DSSS is a modulation technique that enables the receiver to remove undue interference and recover the correct information from an ag-

gregate of multiple signals even when multiple transmitters send information simultaneously over a single channel. Information recovery is made possible with the use of specially-crafted spreading codes.

4.2 Model and Security Definitions

In this section, we briefly describe the algorithms that make up GSPR, and review the security properties of GSPR.

Definition 4.1. *Group Signatures with Probabilistic Revocation*

GSPR is composed of the following algorithms.

- $\mathsf{Setup}(\lambda)$: With the security parameter, $\lambda \in \mathbb{N}$, as the input, this algorithm outputs a group public key $\mathtt{gpk}$, and an issuer's secret key $\mathtt{isk}$. Here, $\mathbb{N}$ represents the set of natural numbers.

- $\mathsf{Join}(\mathtt{isk}, i, m)$: To add the platform $i \in [1, n]$, where n is the total number of platforms in the network, as a member of the group with the secret $\mathtt{isk}$, this algorithm outputs a set of m alias tokens, x_{ik}, $\forall k \in [1, m]$, a corresponding secret/private key $\mathtt{psk}_i$ and a corresponding revocation token $\mathtt{grt}_i$, and makes an entry into a registration list $\mathtt{reg}_i$. In this dissertation, we use the terms "secret key" and "private key" interchangeably.

- $\mathsf{Sign}(\mathtt{gpk}, \mathtt{psk}_i, M)$: With the group public key $\mathtt{gpk}$, and the platform's secret key $\mathtt{psk}_i$, this algorithm outputs a signature σ with alias token x_{ik} on message M.

- $\mathsf{Verify}(\mathtt{gpk}, \mathtt{RC}, \sigma, M)$: If both of the following sub-algorithms produce an output value of $\mathtt{valid}$, this algorithm outputs the value $\boldsymbol{\mathtt{valid}}$; otherwise, it outputs the value $\mathtt{invalid}$.

  - $\mathsf{SignCheck}(\mathtt{gpk}, \sigma, M)$: With the group public key $\mathtt{gpk}$ and a purported signature σ on a message M, this sub-algorithm outputs the value ***valid*** if σ is an honest signature on M; otherwise, it outputs the value `invalid`.
  - $\mathsf{RevCheck}(\mathtt{RC}, \sigma)$: With a revocation code $\mathtt{RC}$ and a purported signature σ, this sub-algorithm outputs the value ***valid*** if the alias token x_{ik} embedded in σ is determined to be unrevoked; otherwise, it outputs the value `invalid`.

- $\mathsf{Revoke}(\mathtt{grt}_i, \mathtt{RC})$: This algorithm updates the revocation code $\mathtt{RC}$ using the revocation token $\mathtt{grt}_i$ if the membership of platform i is to be revoked. Here, revoking the membership of the platform is equivalent to revoking its private key and revoking all of its alias tokens.

- $\mathsf{Open}(\mathtt{reg}, \sigma, M)$: Given a valid signature σ on a message M, created by a platform $i \in [1, n]$, this algorithm outputs the platform's identity i.

In this chapter, we assume that the issuer runs Setup to set-up the group, Join to add a platform to the group, Revoke to revoke a private key of a platform, and Open to open a signature. The platform runs Sign to sign a message, and the verifier runs Verify to verify a signed message.

In the following discussion, we review the three attributes of GSs as per the definitions given in [89].

- *Correctness*: This ensures the following properties.
  - *Signature Correctness*: This ensures that if a signature is generated by an honest platform, the signature check algorithm (i.e., SignCheck) outputs the value ***valid***.
  - *Identity Correctness*: This ensures that if a signature is generated by an honest platform, the issuer correctly reveals the identity of the platform using the signature open algorithm (i.e., Open).

  - *Revocation Correctness*: This ensures that if a signature is generated by an honest platform using an unrevoked private key, the revocation check algorithm (i.e., RevCheck) outputs the value valid.

- *Anonymity*: This property ensures that no party except the issuer is able to identify the platform of a given signature.

- *Traceability*: This property requires that no colluding set of platforms (even consisting of the entire group) can create signatures that cannot be traced back to a platform in the group, or signatures that cannot be traced back to some member of the colluding set.

The revocation correctness property is not considered a core security property in most GSs. However, it is an important property to consider in evaluating our proposed scheme, GSPR, with respect to other GS schemes. GSPR satisfies all of the security properties listed above with the exception of the revocation correctness property. One of the intrinsic attributes of GSPR that distinguishes it from all other GSs is that it satisfies the revocation correctness property with a certain probability, but not with certainty. GSPR exploits the computational efficiency advantage of probabilistic algorithm to significantly reduce the computation cost of the revocation check procedure. Below, we provide formal definitions of the security properties mentioned above.

Definition 4.2. *Signature Correctness*

It requires that for all $\lambda, n \in \mathbb{N}$, all $(\mathtt{gpk}, \mathtt{isk})$ obtained by Setup, all $(\mathtt{psk}_i, \mathtt{grt}_i, \mathtt{reg}_i)$ obtained by Join for any $i \in [1, n]$, and all $M \in \{0,1\}^*$,

$$\mathsf{SignCheck}(\mathtt{gpk}, \mathsf{Sign}(\mathtt{gpk}, \mathtt{psk}_i, M), M) = \mathit{valid}.$$

Definition 4.3. *Identity Correctness*

It requires that for all $\lambda, n \in \mathbb{N}$, all $(\mathtt{gpk}, \mathtt{isk})$ obtained by Setup, all $(\mathtt{psk}_i, \mathtt{grt}_i, \mathtt{reg}_i)$ obtained by Join for any $i \in [1, n]$, and all $M \in \{0,1\}^*$,

$$\mathsf{Open}(\mathtt{reg}, \mathsf{Sign}(\mathtt{gpk}, \mathtt{psk}_i, M), M) = i.$$

Definition 4.4. *Revocation Correctness*

It requires that for all $\lambda, n \in \mathbb{N}$, all $(\mathtt{gpk}, \mathtt{isk})$ obtained by Setup, all $(\mathtt{psk}_i, \mathtt{grt}_i, \mathtt{reg}_i)$ obtained by Join for any $i \in [1, n]$, and all $M \in \{0,1\}^*$,

$$\mathsf{RevCheck}(\mathtt{RC}, \mathsf{Sign}(\mathtt{gpk}, \mathtt{psk}_i, M)) = \mathtt{valid},$$

implies that the private key of the platform i is not revoked.

Definition 4.5. *Anonymity*

It requires that for each PPT algorithm $\mathcal{A}$, the advantage of $\mathcal{A}$ on winning the following game is negligibly small.

1. *Setup*: The challenger runs $\mathsf{Setup}(\lambda)$ and $\mathsf{Join}(\mathtt{isk}, i, m)$, $\forall i \in [1, n]$. He obtains $\mathtt{gpk}$, $\mathtt{psk}$, and $\mathtt{reg}$. He provides the algorithm $\mathcal{A}$ with $\mathtt{gpk}$.
2. *Queries-Phase I*: $\mathcal{A}$ queries the challenger about the following.
   (a) *Signing*: $\mathcal{A}$ requests a signature on an arbitrary message M for an arbitrary member i. The challenger responds with the corresponding signature.
   (b) *Corruption*: $\mathcal{A}$ requests the secret key of an arbitrary member i. The challenger responds with the key $\mathtt{psk}_i$.
   (c) *Opening*: $\mathcal{A}$ requests the identity of the platform by providing a message M and its valid signature σ created by platform $i \in [1, n]$. The challenger responds with the platform's identity i.

3. *Challenge*: $\mathcal{A}$ outputs a message M and two members i_0 and i_1 with the restriction that the corruption of i_0 and i_1 have not been requested. The challenger chooses $\phi \xleftarrow{R} \{0,1\}$, and responds with the signature σ^* on M^* of member i_ϕ.

4. *Queries-Phase II (Restricted Queries)*: After obtaining the challenge, $\mathcal{A}$ can make additional queries of signing, corruption and opening, except the corruption queries of i_0 and i_1, and the opening query of the signature σ^* on M^*.

5. *Output*: $\mathcal{A}$ outputs a bit ϕ' indicating its guess of ϕ.

$\mathcal{A}$ wins the anonymity game if $\phi' = \phi$. The advantage of $\mathcal{A}$ is defined as $|\Pr(\phi' = \phi) - 1/2|$.

Definition 4.6. *Traceability*

It requires that for each PPT algorithm $\mathcal{A}$, the probability that $\mathcal{A}$ wins the following game is negligibly small.

1. *Setup*: The challenger runs $\mathsf{Setup}(\lambda)$ and $\mathsf{Join}(\mathtt{isk}, i, m)$, $\forall i \in [1, n]$. He obtains $\mathtt{gpk}$ $\mathtt{psk}$, and $\mathtt{reg}$. He provides $\mathcal{A}$ with $\mathtt{gpk}$, and sets U as empty.

2. *Queries*: $\mathcal{A}$ queries the challenger about the following.

   (a) *Signing*: $\mathcal{A}$ requests a signature on an arbitrary message M for an arbitrary member i. The challenger responds with the corresponding signature.

   (b) *Corruption*: $\mathcal{A}$ requests the secret key of an arbitrary member i. The challenger adds i to U, and responds with the key $\mathtt{psk}_i$.

3. *Output*: $\mathcal{A}$ outputs a message M^* and a signature σ^*.

$\mathcal{A}$ wins the game if:

1. $\mathsf{SignCheck}(\mathtt{gpk}, \sigma^*, M^*) = \mathtt{valid}$;

2. σ^* is traced to a member outside of U or the trace is failure; and

3. $\mathcal{A}$ did not obtain σ^* by making a signing query on M^*.

4.3 Details of GSPR

In this section, we discuss the technical details of GSPR. For a given security parameter $\lambda \in \mathbb{N}$, we consider a bilinear group pair ($\mathbb{G}_1$,$\mathbb{G}_2$) with isomorphism ψ, and a bilinear mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Note that we utilize either Type-1 or Type-2 bilinear mapping in GSPR as discussed in Section 3.1. We represent $H_z : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{G}_2^2$ as collision resistant hash functions treated as random oracles. Also, we consider a set of alias codes, $\mathbb{C}_p$. The order of $\mathbb{C}_p$ is p which is equal to the order of $\mathbb{Z}_p^*$. Each element in $\mathbb{C}_p$ is an alias code which is a vector of $+1$s and -1s of length l. Further, we define a mapping function $F_c : \mathbb{Z}_p^* \rightarrow \mathbb{C}_p$ using which an alias token in $\mathbb{Z}_p^*$ can be mapped to an alias code in $\mathbb{C}_p$. The details of $\mathbb{C}_p$ and F_c are discussed in Section 4.4.4. $\mathbb{G}_1, \mathbb{G}_2, \psi, H_z, H_2, \mathbb{C}_p$ and F_c are considered public knowledge. In the following paragraphs, we define the algorithms that make up GSPR.

4.3.1 Setup

With the security parameter $\lambda \in \mathbb{N}$ as the input, this algorithm generates the group public key `gpk` and the issuer's secret key `isk` through the following steps.

1. Select a generator $g_2 \xleftarrow{R} \mathbb{G}_2$, and set $g_1 = \psi(g_2)$ such that g_1 is a generator of $\mathbb{G}_1$.
2. Select $\gamma \xleftarrow{R} \mathbb{Z}_p^*$, and compute $w_k = g_2^{\gamma^k}$, $\forall k \in [0, m]$. Note that $w_0 = g_2$.

The group public key is $\texttt{gpk} = (g_1, g_2, w_1, w_2, \cdots w_m)$. The secret belonging only to the issuer is given by $\texttt{isk} = \gamma$. The output of this algorithm is ($\texttt{gpk}, \texttt{isk}$).

4.3.2 Join

This algorithm adds the platform i as a member of the group with the issuer's secret `isk`, and generates m alias tokens for platform i, and a corresponding secret key $\texttt{psk}_i$. This algorithm

also generates a revocation token $\texttt{grt}_i$ for platform i, and an entry in the registration list $\texttt{reg}_i$ using the following steps.

1. Select $y_i \xleftarrow{R} \mathbb{Z}_p^*$.
2. Compute the set of m alias tokens,

$$\texttt{X}_i = \{x_{ik} : x_{ik} = H_z(y_i, k), \forall k \in [1, m]\}, \tag{4.1}$$

   where k^{th} alias token of platform i is represented by x_{ik}.

3. Compute $\pi_i = \prod_{k=1}^{m}(\gamma + x_{ik})$, and calculate

$$A_i = g_1^{1/\pi_i}. \tag{4.2}$$

   In the unlikely case, if $\pi_i = 0$, restart from Step 1.

For platform i, the secret key is $\texttt{psk}_i = (A_i, y_i)$, the revocation token is $\texttt{grt}_i = \texttt{X}_i$, and the entry in the registration list is $\texttt{reg}_i = \texttt{X}_i$. Note that only the issuer has access to $\texttt{reg}$. The output of this algorithm is $(\texttt{psk}_i, \texttt{grt}_i, \texttt{reg}_i)$.

4.3.3 Sign

The inputs to the signing algorithm are the group public key $\texttt{gpk}$, the platform's secret key $\texttt{psk}_i$, and the message to be signed $M \in \{0,1\}^*$. This algorithm generates a signature σ on M using the following steps.

1. Generate the following parameters.
   (a) Compute the alias tokens $\texttt{X}_i$ using equation (4.1).
   (b) Define $\pi_i = \prod_{k=1}^{m}(\gamma + x_{ik}) = \sum_{k=0}^{m} a_k \gamma^k$, where $a_0, a_1, \cdots, a_m \in \mathbb{Z}_p^*$ are the coefficients of the polynomial π_i with the variable γ, and compute

$$B_i = g_2^{\pi_i} = \prod_{k=0}^{m} w_k^{a_k}. \tag{4.3}$$

(c) For each $x_{ik} \in \mathtt{X}_i$, define $\pi_i/(\gamma + x_{ik}) = \prod_{j=1, j\neq k}^{m} (\gamma + x_{ij}) = \sum_{j=0}^{m-1} b_j \gamma^j$, where $b_0, b_1, \cdots, b_{m-1} \in \mathbb{Z}_p^*$ are the coefficients, and compute

$$C_{ik} = g_2^{\pi_i/(\gamma + x_{ik})} = \prod_{j=0}^{m-1} w_j^{b_j}. \tag{4.4}$$

2. Select a tuple $(A_i, B_i, C_{ik}, x_{ik})$ by selecting some value of $k \in [1, m]$. The platform utilizes a particular k to sign all its signatures during some time interval. After this time interval, she discards the alias token. When the platform exhausts all its alias tokens, she runs the Join algorithm again to fetch new secret key, and computes corresponding set of new alias tokens.

3. Compute $(\hat{u}, \hat{v}) = H_2(\mathtt{gpk}, M, x_{ik})$, and calculate their images in $\mathbb{G}_1$, such that $u = \psi(\hat{u})$ and $v = \psi(\hat{v})$.

4. Select $\alpha, \beta, \delta \xleftarrow{R} \mathbb{Z}_p^*$, and compute $T_1 = u^\alpha, T_2 = A_i v^\alpha$, $T_3 = B_i^\beta$, and $T_4 = C_{ik}^\delta$.

5. Compute the proof of knowledge (SPK) which is expressed as follows.

$$\begin{aligned} V = SPK\{(\alpha, \beta, \delta, C_{ik}) : T_1 = u^\alpha, T_2 = A_i v^\alpha, T_3 = B_i^\beta, T_4 = C_{ik}^\delta, e(A_i, B_i) = e(g_1, g_2), \\ e(g_1, B_i) = e(g_1^\gamma g_1^{x_{ik}}, C_{ik})\}(M) \\ = SPK\{(\alpha, \beta, \delta, C_{ik}) : T_1 = u^\alpha, e(T_2, T_3) = e(v, T_3)^\alpha \ e(g_1, g_2)^\beta, \\ 1 = e(g_1, T_3)^\delta \ e(\psi(w_1) g_1^{x_{ik}}, T_4)^{-\beta}\}(M). \end{aligned} \tag{4.5}$$

This SPK is computed with the Fiat-Shamir heuristic method [82] using the following steps.

(a) Select binding factors $r_\alpha, r_\beta, r_\delta \xleftarrow{R} \mathbb{Z}_p^*$, and compute

$$\begin{aligned} R_1 &= u^{r_\alpha}, \\ R_2 &= e(v, T_3)^{r_\alpha} \ e(g_1, g_2)^{r_\beta}, \\ R_3 &= e(g_1, T_3)^{r_\delta} \ e(\psi(w_1) g_1^{x_{ik}}, T_4)^{-r_\beta}. \end{aligned} \tag{4.6}$$

(b) Compute the challenge c as

$$c = H_z(\texttt{gpk}, M, x_{ik}, T_1, T_2, T_3, T_4, R_1, R_2, R_3).$$

(c) Compute responses, $s_\alpha = r_\alpha + c\alpha$, $s_\beta = r_\beta + c\beta$, and $s_\delta = r_\delta + c\delta$.

The output of this algorithm is the signature

$$\sigma = (x_{ik}, T_1, T_2, T_3, T_4, c, s_\alpha, s_\beta, s_\delta). \tag{4.7}$$

4.3.4 Verify

The verification algorithm takes as input the group public key `gpk`, the revocation code `RC`, the signature σ, and the message M. Using the following sub-algorithms, it verifies two things: (1) whether the signature was honestly generated, and (2) revocation status of the alias token used to generate the signature. If both the sub-algorithms output ***valid***, this algorithm outputs ***valid***; otherwise it outputs `invalid`.

- **SignCheck**$(\texttt{gpk}, \sigma, M)$: With the group public key `gpk` and a purported signature σ on a message M, this sub-algorithm outputs ***valid*** if σ is an honest signature on M. This is checked using the following steps.

  1. Compute $(\hat{u}, \hat{v}) = H_2(\texttt{gpk}, M, x_{ik})$, and calculate their images in $\mathbb{G}_1$, i.e., $u = \psi(\hat{u})$ and $v = \psi(\hat{v})$.
  2. Retrieve:

$$\begin{aligned}
\widetilde{R}_1 &= u^{s_\alpha} T_1^{-c}, \\
\widetilde{R}_2 &= e(v, T_3)^{s_\alpha} e(g_1, g_2)^{s_\beta} e(T_2, T_3)^{-c} \\
\widetilde{R}_3 &= e(g_1, T_3)^{s_\delta}\ e(\psi(w_1) g_1^{x_{ik}}, T_4)^{-s_\beta}.
\end{aligned} \tag{4.8}$$

3. Check the correctness of the challenge c as

$$c \stackrel{?}{=} H_z(\texttt{gpk}, M, x_{ik}, T_1, T_2, T_3, T_4, \widetilde{R}_1, \widetilde{R}_2, \widetilde{R}_3).$$

If the above equation holds, this sub-algorithm outputs ***valid***; otherwise, it outputs `invalid`.

- **RevCheck**($\texttt{RC}, \sigma$): The inputs to the revocation check algorithm are the alias token x_{ik} embedded in the signature σ, and the revocation code, `RC`. The purpose of this sub-algorithm is to check whether the alias token, x_{ik}, has been revoked or not, which is accomplished using the following steps.

1. Map x_{ik} to the corresponding alias code s_{ik}, i.e., compute $s_{ik} = F_c(x_{ik})$, where s_{ik} is a column vector of length l of samples of $+1$s and -1s.
2. Compute the value of the decision variable, $z = \frac{1}{l} s_{ik}^T \texttt{RC}$, where s_{ik}^T is the transpose of s_{ik}.
3. Output `invalid` if $z \geq \tau$, where τ is a pre-determined threshold; otherwise, output ***valid***.

4.3.5 Revoke

The inputs to this algorithm are the revocation token of the platform, $\texttt{grt}_i$, and the current revocation code, `RC`. To revoke platform i, the issuer updates the revocation code using the following steps.

1. Map each $x_{ik} \in \texttt{grt}_i$ to the corresponding alias code s_{ik}, i.e., compute $s_{ik} = F_c(x_{ik})$ for $k = 1, 2 \cdots m$.
2. Compute the code, $\bar{s}_i$, by adding all the unique alias codes corresponding to the revoked alias tokens such that $\bar{s}_i = \sum_{k=1}^{m} s_{ik}$.
3. Update the revocation code as $\texttt{RC} = \texttt{RC} + \bar{s}_i$.

4.3.6 Open

With the valid signature σ on message M, the actual signer of the signature is identified using the following step.

1. Search the registration list `reg` to find platform i that has generated signature σ with the alias token x_{ik}.
2. If a match is successfully found, output i; otherwise, output 0 to indicate a failure.

4.4 Security Analysis

4.4.1 Signature and Identity Correctness

It can be shown that GSPR satisfies the signature correctness and the identity correctness properties. Security proofs for these properties can be constructed using the frameworks discussed in [4]. We omit these trivial proofs in this dissertation.

4.4.2 Anonymity

Theorem 4.1. *In the random oracle model, suppose an algorithm $\mathcal{A}$ breaks the anonymity of GSPR with advantage ϵ after q_H hash queries and q_S signing queries, then there exists an algorithm $\mathcal{B}$ that breaks the $\mathbb{G}_2$-DLIN assumption with the advantage $(1/n^2 - q_S q_H/p)\epsilon/2$.*

This theorem prescribes that GSPR satisfies the anonymity property in the random oracle model when the $\mathbb{G}_2$-DLIN assumption (defined in Section 3.4) is presumed. In [4], the core technique used in the proof of anonymity is the randomness of $(\hat{u}, \hat{v})$ such that the challenger can backpatch the hash oracle. GSPR also preserves the randomness of $(\hat{u}, \hat{v})$. Hence, we can employ the same technique, and the proof construction method used in [4] to prove Theorem 4.1. However, here we omit the proof.

Note that within a time interval, the platform uses the same alias token to generate all the signatures, and hence those signatures can be linked to the same platform. However, the platform utilizes different alias tokens in different time intervals, and thus unlinkability of the signatures is preserved between different time intervals. For many applications, the duration of each time interval is small (e.g., 1 minute in vehicular networks [18]), resulting in only a few linkable signatures.

In GSPR, all of the previous signatures generated using a revoked private key can be linked together using the implicit tracing algorithm. The scheme proposed in [4] as well as most other VLR schemes share this drawback. This drawback can be mitigated in a number of ways, including the use of time-stamped parameters [21] or the use of accumulators [90]. However, these methods incur additional overhead that may be unacceptable in many applications.

4.4.3 Traceability

We consider traceability property of GSPR in Theorem 4.3, and utilize Lemma 4.2 to prove it.

Lemma 4.2. *Suppose an algorithm $\mathcal{A}$ which is given an instance $(\tilde{g}_1, \tilde{g}_2, \tilde{g}_2^{\gamma}, \cdots, \tilde{g}_2^{\gamma^m})$ and n tuples $(\tilde{A}_i, x_{i1}, x_{i2}, \cdots x_{im})$, $\forall i \in [1,n]$, where $x_{ik} \in \mathbb{Z}_p^*$ $\forall i \in [1,n], k \in [1,m]$, $\tilde{g}_2 \in \mathbb{G}_2$, $\tilde{g}_1 = \psi(\tilde{g}_2)$ and $\tilde{A}_i = \tilde{g}_1^{1/\prod_{k=1}^{m}(\gamma + x_{ik})}$, forges a tuple $(\tilde{A}_*, \tilde{B}_*, \tilde{C}_*, x_*)$ for some $\tilde{A}_* \in \mathbb{G}_1, \tilde{B}_* \in \mathbb{G}_2, \tilde{C}_* \in \mathbb{G}_2$ and $x_* \neq x_{ik}$ $\forall i \in [1,n], k \in [1,m]$ such that $e(\tilde{A}_*, \tilde{B}_*) = e(\tilde{g}_1, \tilde{g}_2)$ and $e(\tilde{g}_1, \tilde{B}_*) = e(\tilde{g}_1^{\gamma}\tilde{g}_1^{x_*}, \tilde{C}_*)$, then there exists an algorithm $\mathcal{B}$ solves the q-BSDH problem, where $q = (n+1)m$.*

Proof. Algorithm $\mathcal{B}$ is given a q-BSDH instance represented by $(g_1, w_0, w_1, \cdots, w_q)$, where $w_j = g_2^{\gamma^j}$, $\forall j \in [0,q]$. $\mathcal{B}$ sets $q = (n+1)m$. The objective of $\mathcal{B}$ is to produce a BSDH pair $(e(g_1,g_2)^{1/(\gamma+d)}, d)$ for some $d \in \mathbb{Z}_p^*$. For this, $\mathcal{B}$ creates the following framework to interact with $\mathcal{A}$.

1. *Setup*: $\mathcal{B}$ does the following.

   (a) Select nm values: $x_{ik} \xleftarrow{R} \mathbb{Z}_p^*$, $\forall i \in [1,n], k \in [1,m]$.

   (b) Define $\pi_i = \prod_{k=1}^{m}(\gamma + x_{ik})$, and $f(\gamma) = \prod_{i=1}^{n} \pi_i = \sum_{j=0}^{nm} \alpha_j \gamma^j$, where $\alpha_0, \alpha_1, \cdots, \alpha_{nm} \in \mathbb{Z}_p^*$ are the coefficients of the polynomial f with variable γ.

   (c) Compute $\tilde{g_2} = g_2^{f(\gamma)} = \prod_{j=0}^{nm} w_j^{\alpha_j}$, and $\tilde{g_1} = \psi(\tilde{g_2})$.

   (d) Compute $\tilde{w_k} = \tilde{g_2}^{\gamma^k} = \prod_{j=0}^{nm} w_{j+k}^{\alpha_j}$, $\forall k \in [0,m]$.

   (e) Define $f_i(\gamma) = f(\gamma)/\pi_i = \prod_{j=1, j\neq i}^{n} \pi_j = \sum_{j=0}^{nm-m} a_j \gamma^j$, where $a_0, a_1, \cdots, a_{nm-m} \in \mathbb{Z}_p^*$ are the coefficients of the polynomial f_i.

   (f) Calculate $\tilde{D}_i = \tilde{g_2}^{1/\pi_i} = g_2^{f_i(\gamma)} = \prod_{j=0}^{nm-m} w_j^{a_j}$, and $\tilde{A}_i = \psi(\tilde{D}_i)$.

   (g) Send $(\tilde{A}_i, x_{i1}, x_{i2}, \cdots, x_{im})$, $\forall i \in [1,n]$, and $(\tilde{g_1}, \tilde{w_0}, \tilde{w_1}, \cdots, \tilde{w_m})$ to $\mathcal{A}$.

   Note that with this information, $\mathcal{A}$ or $\mathcal{B}$ can compute nm tuples $(\tilde{A}_i, \tilde{B}_i, \tilde{C}_{ik}, x_{ik})$ such that $e(\tilde{A}_i, \tilde{B}_i) = e(\tilde{g_1}, \tilde{g_2})$ and $e(\tilde{g_1}, \tilde{B}_i) = e(\tilde{g_1}^{\gamma} \tilde{g_1}^{x_{ik}}, \tilde{C}_{ik})$ in the following manner.

      i. Define $\pi_i = \prod_{k=1}^{m}(\gamma + x_{ik}) = \sum_{k=0}^{m} b_k \gamma^k$, where $b_0, b_1, \cdots b_m \in \mathbb{Z}_p^*$ are the coefficients of the polynomial defined by π_i.

      ii. Compute $\tilde{B}_i = \tilde{g_2}^{\pi_i} = \prod_{k=0}^{m} \tilde{w_k}^{b_k}$.

      iii. Define $f_{ik}(\gamma) = \pi_i/(\gamma + x_{ik}) = \prod_{j=1, j\neq k}^{m}(\gamma + x_{ij}) = \sum_{j=0}^{m-1} c_j \gamma^j$, where $c_0, c_1, \cdots c_{m-1} \in \mathbb{Z}_p^*$ are the coefficients of the polynomial f_{ik}.

      iv. Compute $\tilde{C}_{ik} = \tilde{g_2}^{f_{ik}(\gamma)} = \prod_{j=0}^{m-1} \tilde{w_j}^{c_j}$.

   Also, $\mathcal{A}$ or $\mathcal{B}$ can compute nm BSDH pairs $(\tilde{E}_{ik}, x_{ik})$ in the following manner.

$$\tilde{E}_{ik} = e(\tilde{A}_i, \tilde{C}_{ik}) = e(\tilde{g_1}, \tilde{g_2})^{1/(\gamma + x_{ik})}.$$

2. *Output*: $\mathcal{A}$ outputs a forged tuple $(\tilde{A}_*, \tilde{B}_*, \tilde{C}_*, x_*)$, for some $\tilde{A}_* \in \mathbb{G}_1, \tilde{B}_* \in \mathbb{G}_2, \tilde{C}_* \in \mathbb{G}_2$ and $x_* \neq x_{ik}$, $\forall i \in [1,n], k \in [1,m]$, such that $e(\tilde{A}_*, \tilde{B}_*) = e(\tilde{g_1}, \tilde{g_2})$ and $e(\tilde{g_1}, \tilde{B}_*) = e(\tilde{g_1}^{\gamma} \tilde{g_1}^{x_*}, \tilde{C}_*)$.

Having received the forged tuple from $\mathcal{A}$, $\mathcal{B}$ generates a new BSDH pair in the following manner.

1. Define

$$E' = e(A_*, C_*) = e(\tilde{g}_1, \tilde{g}_2)^{1/(\gamma+x_*)} = e(\tilde{g}_1, g_2)^{f(\gamma)/(\gamma+x_*)}.$$

2. Rewrite $f(\gamma)$ as $f(\gamma) = (\gamma + x_*)f_d(\gamma) + d_*$ for some polynomial $f_d(\gamma) = \sum_{j=0}^{nm-1} d_j\gamma^j$, and constant $d_* \in \mathbb{Z}_p^*$. This means that

$$E' = e(\tilde{g}_1, g_2)^{f_d(\gamma)+d_*/(\gamma+x_*)}.$$

3. Compute $g_2^{f_d(\gamma)} = \prod_{j=0}^{nm-1} w_j^{d_j}$, and

$$\tilde{E} = \left(E'/e(\tilde{g}_1, g_2^{f_d(\gamma)})\right)^{1/d_*} = e(\tilde{g}_1, g_2)^{1/(\gamma+x_*)} = e(g_1, g_2)^{f(\gamma)/(\gamma+x_*)} = e(g_1, g_2)^{f_d(\gamma)+d_*/(\gamma+x_*)}.$$

4. Calculate

$$E_* = \left(\tilde{E}/e(g_1, g_2^{f_d(\gamma)})\right)^{1/d_*} = e(g_1, g_2)^{1/(\gamma+x_*)}.$$

Hence, $\mathcal{B}$ returns the tuple (E_*, x_*) as the solution to the submitted instance of the BSDH problem.

□

Theorem 4.3. *In the random oracle model, suppose an algorithm $\mathcal{A}$ breaks the traceability of GSPR with advantage ϵ, after q_H hash queries and q_S signature queries, then there exists an algorithm $\mathcal{B}$ that breaks the q-BSDH assumption with advantage $(\epsilon/n-1/p)^2/16q_H$, where $q = (n+1)m$.*

Proof. The following is an interaction between $\mathcal{A}$ and $\mathcal{B}$.

1. *Setup*: $\mathcal{B}$ is given a bilinear group pair $(\mathbb{G}_1, \mathbb{G}_2)$ with respective generators g_1 and g_2. $\mathcal{B}$ is also given $(w_0, w_1, \cdots w_m)$, where $w_k = g_2^{\gamma^k}$, $\forall k \in [0, m]$. Further, $\mathcal{B}$ is given (A_i, y_i), $\forall i \in [1, n]$. For each i, either $s_i = 1$ indicating that a valid key pair (A_i, y_i) generated using equations (4.1) and (4.2) is known, or $s_i = 0$ indicating that A_i corresponding to y_i is not known. We run $\mathcal{A}$ giving it $\mathsf{gpk} = (g_1, w_0, w_1, \cdots w_m)$ and y_i, $\forall i \in [1, n]$. Note that y_i can be used to generate the alias tokens using equation (4.1).

2. *Queries*: $\mathcal{A}$ can query $\mathcal{B}$ about the following.

   (a) *Hash queries*: $\mathcal{A}$ queries the hash functions H_z and H_2, and B responds with random values with consistency.

   (b) *Signing queries*: $\mathcal{A}$ requests a signature of member i on message M. If $s_i = 1$, $\mathcal{B}$ responds with the signature σ using Sign algorithm with the private key (A_i, y_i). If $s_i = 0$, $\mathcal{B}$ selects $x_{ik}, \alpha, \beta, \delta$ to compute T_1, T_2, T_3 and T_4 and the SPK V as in equation (4.5). If the hash function causes a collision, $\mathcal{B}$ declares failure and aborts; otherwise, $\mathcal{B}$ responds with $\sigma = (x_{ik}, T_1, T_2, T_3, T_4, c, s_\alpha, s_\beta, s_\delta)$. We assume that the signing queries related to a platform does not exceed m.

   (c) *Corruption queries*: $\mathcal{A}$ requests the secret key of member i. If $s_i = 1$, $\mathcal{B}$ adds i to U, and responds with (A_i, y_i); otherwise, $\mathcal{B}$ declares failure and aborts. With (A_i, y_i), $\mathcal{A}$ can compute alias tokens x_{ik}, $\forall k \in [1, m]$ using equation (4.1), B_i using equation (4.3), and C_{ik}, $\forall k \in [1, m]$ using equation (4.4).

3. *Output*: Finally, if $\mathcal{A}$ is successful, it outputs a forged signature σ^* on a message M^* using tuple $(A_{i'}, B_{i'}, C_{i'k}, x_{i'k})$. If $\mathcal{B}$ fails to find the platform i' in U, it outputs σ^*; otherwise, $\mathcal{B}$ identifies some $i' = i$. If $s_{i'} = 0$, $\mathcal{B}$ outputs σ^*; otherwise, $\mathcal{B}$ declares failure and aborts.

With the above framework, there can be two types of forger algorithms [4]. Type I forger forges a signature of the member who is different from all $i \in [1, n]$. Type II forger forges a signature of the member $i \in [1, n]$ whose corruption is not requested. $\mathcal{B}$ treats these two

types of forgers differently. Note that using the technique of Lemma 4.2, with a q-BSDH instance $(\hat{g}_1, \hat{g}_2, \hat{g}_2^{\gamma}, \cdots \hat{g}_2^{\gamma^q})$, $\mathcal{B}$ can obtain $(g_1, g_2, g_2^{\gamma^k}, \cdots, g_2^{\gamma^m})$, and $(q-m)$ BSDH pairs. Moreover, any BSDH pair besides these $(q-m)$ pairs can be transformed into a solution to the original q-BSDH instance which means that the q-BSDH assumption is broken.

Type I Forger: From an instance of $(n+1)m$-BSDH, $\mathcal{B}$ obtains $(g_1, g_2, g_2^{\gamma^k}, \cdots, g_2^{\gamma^m})$, and n tuples $(A_i, x_{i1}, x_{i2}, \cdots x_{im})$. From these n tuples, $\mathcal{B}$ obtains n valid key pairs (A_i, y_i) by setting $H_z(y_i, k) = x_{ik}$, $\forall i \in [1, n], k \in [1, m]$. $\mathcal{B}$ applies the above framework to $\mathcal{A}$. The framework succeeds whenever $\mathcal{A}$ succeeds. Hence, $\mathcal{B}$ obtains the Type I forgery with the probability ϵ.

Type II Forger: From an instance of nm-BSDH, $\mathcal{B}$ obtains $(g_1, g_2, g_2^{\gamma^k}, \cdots, g_2^{\gamma^m})$, and $n-1$ tuples $(A_i, x_{i1}, x_{i2}, \cdots x_{im})$. From these $n-1$ tuples, $\mathcal{B}$ obtains $n-1$ valid key pairs (A_i, y_i) by setting $H_z(y_i, k) = x_{ik}$ $\forall i \in [1, n-1], k \in [1, m]$. These $n-1$ pairs (A_i, y_i) are distributed among n indices. $\mathcal{B}$ sets $s_{i'} = 0$ for the unfilled entry at random index i'. $\mathcal{B}$ selects $A_{i'} \xleftarrow{R} \mathbb{G}_1$, and $y_i \xleftarrow{R} \mathbb{Z}_p^*$. $\mathcal{B}$ applies the framework to $\mathcal{A}$. The framework succeeds only if $\mathcal{A}$ never requests the corruption of member i', but forges a signature that traces to $A_{i'}$. The value of i' is independent of the views of $\mathcal{A}$, and hence $\mathcal{B}$ obtains the Type II forgery with probability at least ϵ/n.

$\mathcal{B}$ obtains another BSDH pair beyond the given nm BSDH pairs using the framework with Type I or Type II forger in the following manner, contradicting the BSDH assumption. $\mathcal{B}$ rewinds the framework to obtain two forged signatures on the same message, where the commitments are the same, but the challenges and responses are different. The probability of success in achieving this is at least $(\epsilon' - 1/p)^2/16q_H$ by the forking lemma [4, 91], where ϵ' is the probability that the framework on each forger succeeds. $\mathcal{B}$ extracts (A_*, B_*, C_*, x_*) encoded in the forged signatures [4]. Further, $\mathcal{B}$ obtains a BSDH pair from (A_*, B_*, C_*, x_*) using the technique discussed in Lemma 4.2. The framework is successful only if the extracted BSDH pair is not among the BSDH pairs created by $\mathcal{B}$. Therefore, $\mathcal{B}$ obtains a new BSDH pair with the probability $(\epsilon' - 1/p)^2/16q_H$.

Hence, we have shown that $\mathcal{B}$ can solve the $(n+1)m$-BSDH instance with probability $(\epsilon - 1/p)^2/16q_H$ using Type I forger, and the nm-BSDH instance with probability $(\epsilon/n - 1/p)^2/16q_H$ using Type II forger. Therefore, the pessimistic Type II forger proves the theorem. This implies that traceability is satisfied in GSPR in the random oracle model under the BSDH assumption.

□

4.4.4 Revocation Correctness

In the following discussion, we analyze the correctness of the results generated by the revocation check algorithm, RevCheck. The revocation correctness depends on the cross correlation property of the alias codes since the revocation code is generated by summing over multiple alias codes. Here, we discuss two categories of codes from the existing literature which can be potentially used as alias codes—orthogonal codes and non-return-to-zero (NRZ) based random codes. Through analytical results, we show that orthogonal codes and random codes are both inadequate for use in GSPR. Hence, we propose a new type of codes which we refer to as *piecewise-orthogonal codes* which can be used as alias codes. With the use of piecewise-orthogonal codes, GSPR's RevCheck algorithm does not determine the revocation status of a private key with certainty, but instead with a certain probability. If an alias token has been revoked and its corresponding alias code has been included in the revocation code, then RevCheck's result is guaranteed to be correct. However, there is a possibility of a false alarm. Using an iterative algorithm, this probability can be decreased iteratively, a la the well-known Miller-Rabin primality test algorithm [92].

For analyzing the revocation correctness, we define the two hypotheses—$H_0 : x_{ik}$ has been revoked, and $H_1 : x_{ik}$ has not been revoked. Here, the probability of false negative/dismissal, P_{fd}, can be defined as the probability of erroneously determining that a given alias token has not been revoked when it has been revoked by the issuer. In RevCheck, P_{fd} is equal to the probability of $z < \tau$ when H_0 is true. Also, probability of false positive/alarm, P_{fa}, can

be defined as the probability of the verifier erroneously determining that a given alias token has been revoked when it has not been revoked by the issuer. In RevCheck, P_{fa} is equal to the probability of $z \geq \tau$ when H_1 is true. Further, we suppose that the number of revoked private keys is represented by n_r, and each alias token (or each element in $\mathbb{Z}_p^*$) is represented by $b_p = 160$ bits. Note that the number of revoked alias tokens (i.e., $m \cdot n_r$) is equal to the number of revoked alias codes, and the length of the revocation code is equal to the length of an alias code (i.e., l).

Orthogonal Codes: Orthogonal or Walsh codes consist of codes with zero cross-correlation [93]. When the two codes are the same, the value of the cross-correlation is 1; otherwise, it is 0. If these codes are used as alias codes, we can set the threshold $\tau = 1$, and the revocation check procedure with $P_{fd} = 0$ and $P_{fa} = 0$ can be achieved. This means that if we use orthogonal codes as alias codes, GSPR would be able to satisfy the revocation correctness property with certainty. However, there are only l unique orthogonal codes of length l samples. This means that if orthogonal codes are indexed using the alias token x_{ik} which is represented by $b_p = 160$ bits, then the length of each alias code has to be $l = 2^{160}$ samples long! Hence, it is prohibitively costly in terms of storage and processing overhead to use completely orthogonal codes as alias codes.

NRZ based Random Codes: Random codes can be generated by NRZ encoding of a random sequence of bits, which means bit 0 is mapped to sample -1, and bit 1 is mapped to sample $+1$. As a result, the number of unique random codes with length b_p is given by 2^{b_p}. If the random codes are utilized as alias codes, we can generate an alias code of length, $l = b_p = 160$ samples, by NRZ encoding of an alias token, x_{ik}. Although the use of random codes (as alias codes) allows us to use compact alias codes, they have a critical drawback; use of random codes results in $P_{fd} > 0$. As a result of the random nature of the codes, there are inevitable false dismissals, which means there is significant possibility that the verifier would not be able to detect a revoked private key. This is untenable in PPA as this could be utilized by adversaries to bypass the revocation check.

As discussed above, orthogonal codes as well as random codes have critical drawbacks that limit their utility as alias codes. Hence, we propose a new type of codes that we call *piecewise-orthogonal codes*. The use of piecewise-orthogonal codes enables us to create alias codes that are compact and have a very desirable property—viz., $P_{fd} = 0$ and $P_{fa} > 0$. In other words, when we use piecewise-orthogonal codes, the probability of false dismissals is guaranteed to be zero, although the probability of false alarms is non-zero. Note that ensuring $P_{fd} = 0$ is much more important than $P_{fa} = 0$ from a security point of view. The former implies that a revoked alias token can detected by RevCheck with 100% certainty. In the next subsection, we provide details on how piecewise- orthogonal codes are used in probabilistic revocation.

Revocation with Piecewise-Orthogonal Codes

In GSPR, we utilize piecewise-orthogonal codes as alias codes for achieving probabilistic revocation. The piecewise-orthogonal codes are generated by concatenating multiple segments where each segment is an orthogonal code. To generate a piecewise-orthogonal code as an alias code, an alias token is divided into multiple segments, and an orthogonal code is generated corresponding to each segment. These orthogonal codes corresponding to the segments of the alias token are concatenated to form the complete alias code. In this way, the alias codes are piecewise-orthogonal.

Specifically, a set of 2^{b_s} orthogonal codes, denoted by $\mathbb{C}_s$, is generated using the technique discussed in [93], where each orthogonal code is of length 2^{b_s}. Note that an orthogonal code in $\mathbb{C}_s$ can be retrieved using a b_s-bit index. Further, each alias token $x_{ik} \in \mathbb{Z}_p^*$ of b_p bits is divided into d segments each of length b_s bits, such that $d \cdot b_s \leq b_p < (d+1) \cdot b_s$. The segments of the alias token x_{ik} are represented by $x_{ik,j}$, $\forall j \in [1, d]$. Further, $\forall j \in [1, d]$, $x_{ik,j}$ is utilized to generate b_s-bit index so that an orthogonal code $s_{ik,j}$ is chosen from $\mathbb{C}_s$. Finally, all the d orthogonal codes, $s_{ik,j}$, $\forall j \in [1, d]$, are concatenated to generate the alias code s_{ik}. The length of the resulting revocation code is $l = d \cdot 2^{b_s}$. The issuer declares the two public parameters $\mathbb{C}_p$ and F_c, such that the set of all possible alias codes $\mathbb{C}_p = \mathbb{C}_s^d$, and

the mapping function $F_c : \mathbb{Z}_p^* \rightarrow \mathbb{C}_p$ is defined as segment-wise indexing as discussed above.

When the revocation code is generated using the Revoke algorithm, each segment of the revocation code is generated by summation of the corresponding segments of the revoked alias codes. Hence, the generated revocation code also has d segments, represented by $\texttt{RC}_j$, $\forall j \in [1, d]$. Note that due to the property of orthogonal codes, the cross-correlation of a revocation code's segment and an orthogonal code results in one of the two values—(1) 0 if the revocation code was not generated by the orthogonal code, or (2) an integral multiple of 1 if the revocation code was generated by the orthogonal code. Hence, the threshold τ is set to 1.

Having received a signature with alias token x_{ik}, the verifier can run RevCheck for each of the d segments. However, to minimize the computational overhead, the verifier only runs RevCheck for a segments. This means that RevCheck can be re-organized as follows.

RevCheck$(\texttt{RC}, \sigma)$

1. Set $j = 1$.
2. Generate a b_s-bit index from $x_{ik,j}$, and select an orthogonal code $s_{ik,j}$ from $\mathbb{C}_s$.
3. Compute $z = \frac{1}{2^{b_s}} s_{ik,j}^T \cdot \texttt{RC}_j$. If $z \geq 1$, output invalid; otherwise, output `valid`, and exit.
4. Set $j = j + 1$. If $j \leq a$, go to Step 2; otherwise, exit.

Example

We illustrate the revocation check procedure in GSPR through an example. The alias codes and the revocation code used in the example are given in Table 4.1. We assume that there are five 4-bit alias tokens represented by $x_1 = \{1111\}$, $x_2 = \{1010\}$, $x_3 = \{0101\}$, $x_4 = \{1101\}$ and $x_5 = \{1110\}$. Also, we assume that $\mathbb{C}_s$ contains $2^2 = 4$ orthogonal codes. The issuer generates the alias codes s_1, s_2, s_3, s_4, and s_5—corresponding to x_1, x_2, x_3, x_4, and x_5,

Table 4.1: The alias and revocation codes used in the illustration example of GSPR.

s_1	$+1$	-1	-1	$+1$	$+1$	-1	-1	$+1$
s_2	$+1$	$+1$	-1	-1	$+1$	$+1$	-1	-1
s_3	$+1$	-1	$+1$	-1	$+1$	-1	$+1$	-1
s_4	$+1$	-1	-1	$+1$	$+1$	-1	$+1$	-1
s_5	$+1$	-1	-1	$+1$	$+1$	$+1$	-1	-1
$\mathtt{RC} = s_1 + s_2$	$+2$	0	-2	0	$+2$	0	-2	0

respectively—by concatenating two orthogonal codes of length 4 samples. Suppose that the issuer needs to revoke alias tokens x_1 and x_2. Hence, the issuer computes the sample-by-sample addition of the alias codes s_1 and s_2. The resulting vector is the revocation code, represented by $\mathtt{RC}$. The issuer provides the verifier with $\mathtt{RC}$. In this scenario, if the verifier receives a signature with the alias token x_1, he runs two iterations of RevCheck. In the first iteration, the verifier computes the cross correlation between the first segments, i.e., first 4 samples of s_1 and $\mathtt{RC}$, represented by $s_{1,1}$ and $\mathtt{RC}_1$, respectively. In the second iteration, the verifier computes the cross correlation between the second segments, i.e., second 4 samples of s_1 and $\mathtt{RC}$ represented by $s_{1,2}$ and $\mathtt{RC}_2$, respectively. The cross correlation is computed by sample-by-sample multiplication of the alias code and the revocation code followed by the addition of all the products, and the resulting value is given by $\frac{1}{4}s_{1,1}^T \cdot \mathtt{RC}_1 = 1$, and $\frac{1}{4}s_{1,2}^T \cdot \mathtt{RC}_2 = 1$. Since the cross correlation of both the segments resulted in the value of 1, the verifier concludes that x_1 has been revoked. Using the same procedure, the verifier concludes that x_2 has also been revoked. On the other hand, if the verifier receives a signature with the alias token x_3, he will conclude that the alias token is valid because the cross correlation of $s_{3,1}$ with $\mathtt{RC}_1$ is 0. Here, the RevCheck algorithm exits after the first iteration. Now, let us take a look at x_4. The cross correlation of $s_{4,1}$ and $s_{4,2}$ with $\mathtt{RC}_1$ and $\mathtt{RC}_2$ results in the values 1 and 0, respectively. Here, if the verifier makes a decision after only computing the correlation of the first segment, to decrease its computational overhead, he erroneously determines that x_4 has been revoked because this is an instance of a false alarm. However, after computing

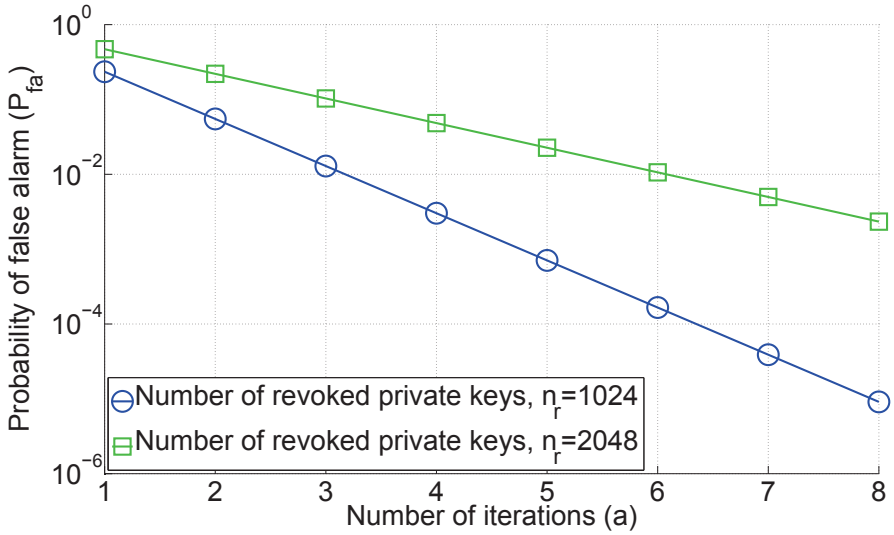


Figure 4.1: Probability of false alarm vs. number of iterations in GSPR.

the correlation of the second segment, the verifier can conclude with absolute confidence that x_4 has not been revoked because $P_{fd} = 0$. Lastly, the cross correlation of $s_{5,1}$ and $s_{5,2}$ with $\texttt{RC}_1$ and $\texttt{RC}_2$ results in 1 and 1, respectively. Hence, if the verifier receives a signature with an alias token x_5, he erroneously concludes that x_5 has been revoked.

Discussions on the False Alarm Probability

With the proposed piecewise-orthogonal codes, the probability of false dismissal is zero, i.e., $P_{fd} = 0$. However, after checking a segments, the upper bound of the probability of false alarm (P_{fa}) can be computed to be

$$P_{fa} = \frac{(mn_r)^a \; 2^{b_p - ab_s} - mn_r}{2^{b_p} - mn_r} = \frac{n_t^a \; 2^{b_p} - mn_r}{2^{b_p} - mn_r} \approx n_t^a \tag{4.9}$$

where the ratio of the number of revoked alias tokens and the length of one segment of the revocation code is represented by $n_t = mn_r/2^{b_s}$. For $n_t < 1$, P_{fa} decreases by increasing a which is the maximum number of iterations or segments processed by the verifier before making a revocation status decision. Here, we assume that all revoked alias tokens have unique segments, and hence the above equation gives the upper bound of P_{fa}. Note that

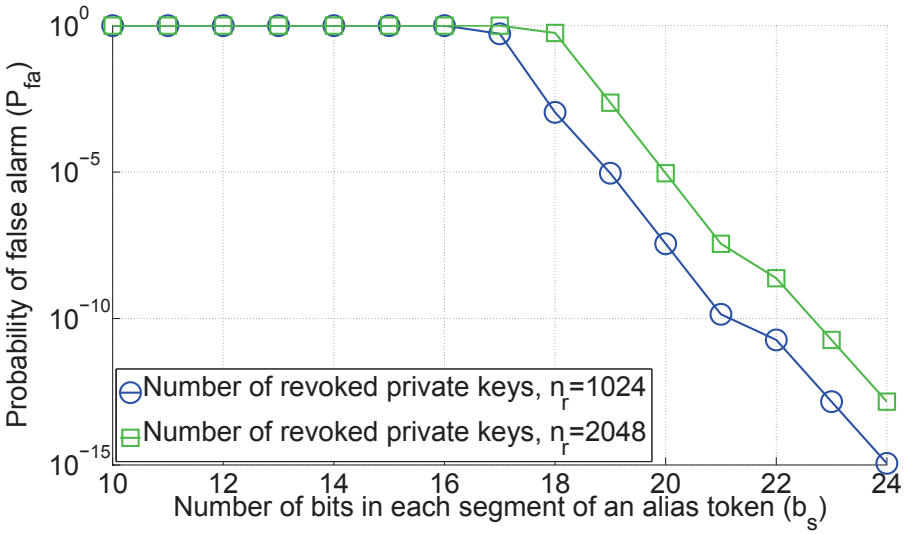


Figure 4.2: Probability of false alarm vs. number of bits in each segment of an alias token in GSPR.

each alias token of b_p bits is unique; however each segment of an alias token, which is b_s bits long, is not necessarily unique.

If the verifier runs RevCheck for a iterations, then the length of the alias code that has to be processed is $l_r = a \cdot 2^{b_s}$, and $P_{fa} \approx n_t^{l_r n_t / m n_r}$. Note that the computational overhead for RevCheck is directly proportional to l_r. There is a tradeoff between P_{fa} and RevCheck's computational cost, and there are a number of different strategies for making an advantageous tradeoff. One possible strategy is to construct the revocation code in such a manner that minimizes P_{fa} for a given value of l_r and for a given number of revoked alias tokens (i.e., mn_r) by selecting an optimal value of b_s. Once the optimal value of b_s is computed, the corresponding n_t can be computed using the relation $n_t = mn_r/2^{b_s}$. This value can be readily derived as $n_t = \exp(-1) \approx 0.3679$. However, mn_r and 2^{b_s} are both integer values, and hence to minimize P_{fa}, the issuer needs to select b_s such that $\exp(-1)/2 \leq mn_r/2^{b_s} < 3\exp(-1)/2$.

As discussed above, the number of iterations (i.e., a) and the number of bits in each segment of an alias token (i.e., b_s) are adjustable parameters that directly impact P_{fa}. Figure 4.1 shows the impact of a on P_{fa} for a fixed value of $b_s = 19$. This figure suggests that the verifier

can decrease P_{fa} at the cost of increasing the computational cost of performing RevCheck. Figure 4.2 illustrates the impact of b_s on P_{fa} when the verifier utilizes all of the d segments of the revocation code to check the revocation status of an alias token. In both figures, we fixed the values $m = 120$ and $b_p = 160$ bits.

Security Implications of the Alias Codes

There is a one-to-one mapping between an alias code and an alias token defined by F_c. Although the alias codes have a non-random structure, the alias tokens, which are embedded in the signature, are random numbers under the random oracle model. Hence, the use of alias codes should have no impact on the traceability property of GSPR, which is defined by Theorem 4.3.

4.5 Performance Evaluation

In this section, we evaluate the computational and communication overhead of GSPR, and compare GSPR's performance with two schemes in the prior art—the Boneh-Shacham (BS) scheme proposed in [4] and the Bichsel-Camenisch-Neven-Smart-Warinschi (BCNSW) scheme proposed in [2]. In [94], Manulis et al. concluded that BS and BCNSW are two of the most practical group signature schemes in terms of being scalable to large networks. We assume that isomorphism is an identity map which means that $\mathbb{G}_1 = \mathbb{G}_2$. We assume symmetric 80-bit security level, which provides approximately the same level of security as an RSA signature with a modulus size of 1024 bits. In an elliptic curve cryptosystem, to achieve the same security strength, the length of an element in $\mathbb{Z}_p^*$, and $\mathbb{G}_1$ needs to be approximately equal to 160 bits [4]. Specifically, we utilize the "Type A" internal described in pairing-based cryptography (PBC) library available at [95]. The internal is constructed on a supersingular curve of the form $y^2 = x^3 + x$ over the field F_q for some prime $q = 3 \bmod 4$. In the internal, an element in $\mathbb{Z}_p^*$ is denoted by 160 bits, and an element in $\mathbb{G}_1$ or $\mathbb{G}_2$ is

Table 4.2: Comparison of computationally expensive operations in GS schemes.

		Exp. in $\mathbb{G}_1/\mathbb{G}_2$	Exp. in $\mathbb{G}_T$	Bilinear mapping
GSPR	Sign	6	4	3
	SignCheck	2	5	4
	RevCheck	0	0	0
BS	Sign	5	3	3
	SignCheck	4	4	4
	RevCheck	0	0	$n_r + 1$
BCNSW	Sign	3	1	1
	SignCheck	0	2	5
	RevCheck	0	0	n_r

denoted by 512 bits. For GSPR, we assume that the issuer distributes 120 alias tokens for each platform, and the verifier needs the probability of false alarm to be less than 0.01.

4.5.1 Computational Overhead

In this section, we compare the computational cost of GSPR with two benchmarks—viz., BS and BCNSW. We focus on three specific algorithms: Sign (signature generation algorithm), SignCheck (signature correctness checking algorithm), and RevCheck (revocation status checking algorithm). We focus on those algorithms because they need to be executed on-line in real time, and moreover they need to be performed by the platform and the verifier, who have limited computational capabilities compared to the issuer.

Firstly, we consider only the most computationally expensive operations—i.e., exponentiation (Exp.) in $\mathbb{G}_1$, $\mathbb{G}_2$, or $\mathbb{G}_T$, and bilinear mapping. Here, since $\mathbb{G}_1 = \mathbb{G}_2$, the application of isomorphism is not considered. Table 4.2 provides the number of operations needed in each of the three algorithms for GSPR, BS and BCNSW. Note that in GSPR, the opera-

Table 4.3: Comparison of computational overhead (in ms) in GS schemes.

	Sign	SignCheck	RevCheck
GSPR	14.952	9.124	5.819
BS	15.417	15.378	1628.729
BCNSW	3.242	8.302	1592.019

tions in Step 1 in the Sign algorithm are independent of the message to be signed or the random parameters, and hence, they can be pre-computed. Also, $\psi(w_1)$ and $e(g_1, g_2)$ can also be pre-computed. Further, in the RevCheck algorithm in GSPR, the computational cost of computing the cross-correlation between a revocation code and an alias code is l integer additions since the length of the revocation code is l with each element being an integer, and the alias code is a vector of $+1$s and -1s.

By using the PBC library, we implement the three algorithms for GSPR, BS and BCNSW, and measure their running time on a PC platform with Intel(R) Core(TM)2 Duo CPU E8400 @ 3GHz. The measurements are obtained by averaging over 1000 runs of each algorithm. Table 4.3 provides their running times on the PC platform. Here, we assume that the number of revoked private keys is 1024, i.e, $n_r = 1024$. From Table 4.3, we can observe that there is no significant difference in the computation times of the three schemes when comparing their performance with respect to Sign and SignCheck. However, the difference between GSPR and the other two schemes in terms of the computational cost of RevCheck is significant. GSPR's RevCheck algorithm is more than two orders of magnitude more efficient than those of the other two schemes. Hence, when we consider the total signature verification time which includes the time needed to perform SignCheck as well as RevCheck, the running time in GSPR is significantly less than that in BS and BCNSW.

Figure 4.3 shows the computation time required to verify a signature versus the number of revoked private keys. We observe that with only a few thousand revoked private keys, the computation times for BS and BCNSW quickly grow to several seconds for verifying only one

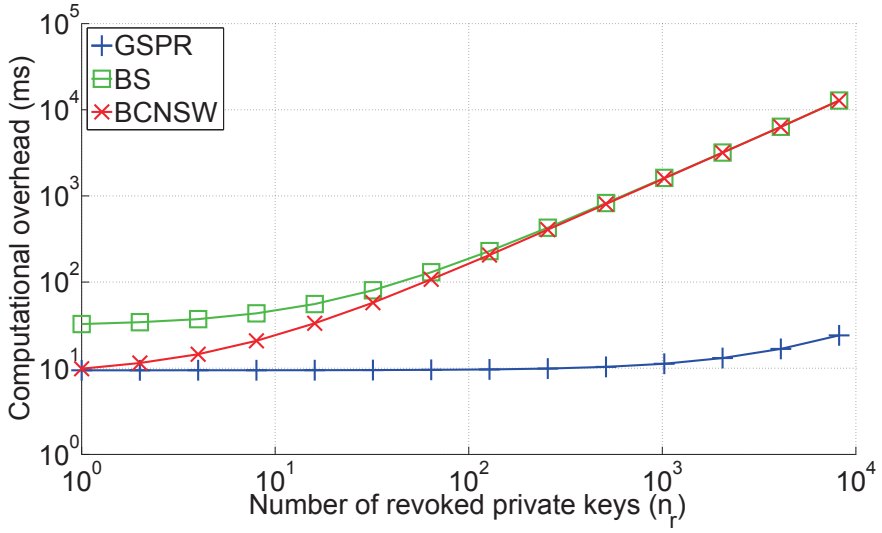


Figure 4.3: Comparison of computational overhead of verifying a signature in GSs vs. the number of revoked private keys.

signature. In contrast, the growth rate of GSPR's computation time is much lower, which is primarily due to the computational efficiency advantage of GSPR's RevCheck.

4.5.2 Communication Overhead

We consider the three communication scenarios—between the issuer and the platform (issuer-platform), between the platform and the verifier (platform-verifier), and between the issuer and the verifier (issuer-verifier). In the first scenario, while joining the group, the issuer sends a secret key to the platform. In the second scenario, the platform sends a signature to the verifier. Lastly, in the third scenario, the issuer sends a revocation list/code to the verifier. Table 4.4 provides the number of elements (Elem.) of $\mathbb{Z}_p^*$, $\mathbb{G}_1$, $\mathbb{G}_2$ or integers (Int.) communicated in each of the three scenarios for GSPR, BS and BCNSW. Note that in GSPR, the alias tokens are generated by the platform using the secret key obtained from the issuer, and hence they do not need to be communicated.

Table 4.5 shows the required communication overhead of the three schemes for the three

Table 4.4: Comparison of number of elements communicated in the considered scenarios in GS schemes.

		Elem. in $\mathbb{Z}_p^*$	Elem. in $\mathbb{G}_1/\mathbb{G}_2$	Int.
GSPR	issuer-platform	1	1	0
	platform-verifier	5	4	0
	issuer-verifier	0	0	l
BS	issuer-platform	1	1	0
	platform-verifier	5	2	0
	issuer-verifier	0	n_r	0
BCNSW	issuer-platform	1	3	0
	platform-verifier	2	3	0
	issuer-verifier	0	n_r	0

Table 4.5: Comparison of communication overhead (bits) in GS schemes.

	issuer-platform	platform-verifier	issuer-verifier
GSPR	672	2848	$5.03 \cdot 10^7$
BS	672	1824	$5.24 \cdot 10^5$
BCNSW	1696	1856	$5.24 \cdot 10^5$

scenarios, assuming $n_r = 1024$. Results from the table indicate that GSPR's communication overhead is two orders of magnitude larger than those of the other two schemes when considering the issuer-verifier scenario. Hence, we can conclude that GSPR makes an advantageous trade-off between computational overhead and communication overhead. This trade-off is advantageous because reducing the computational overhead is much more critical than reducing the communication overhead when considering scalability. Verifying a signature (which includes checking the revocation status of the private key) is an inherently on-line task which needs to be performed in real time, and it can be the primary performance bottleneck when the scheme is deployed in a large network. However, the greater

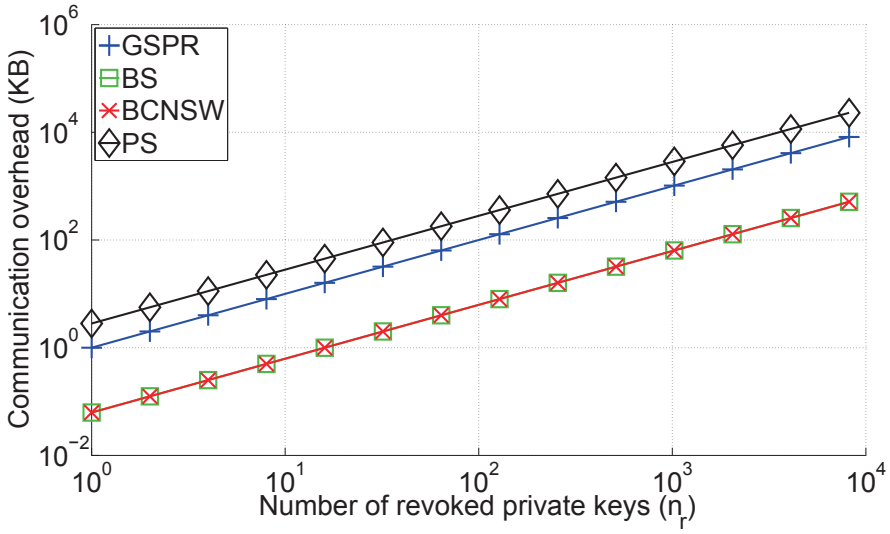


Figure 4.4: Comparison of the communication overhead of transmitting the revocation list/code in GSs vs. the number of revoked private keys.

communication overhead incurred by GSPR in the third (i.e., issuer-verifier) scenario can be readily mitigated by pre-fetching the revocation code before the verifier needs to verify a given signature.

In Figure 4.4, we compare four schemes in terms of the communication overhead required to transmit the revocation list (for GSPR, it is the revocation code). The top-most curve is the curve for a pseudonym-based signature (PS) with Elliptic Curve Digital Signature Algorithm (ECDSA) with the public key size of 192 bits to achieve the 80-bit security level. For PS, we assume that the number of pseudonyms allotted to each platform is 120, and the issuer publishes public-key certificates of all the revoked pseudonyms in the revocation list. In Figure 4.4, we observe that although the communication overhead of GSPR is higher as compared to BS and BCNSW, it is still lower than PS.

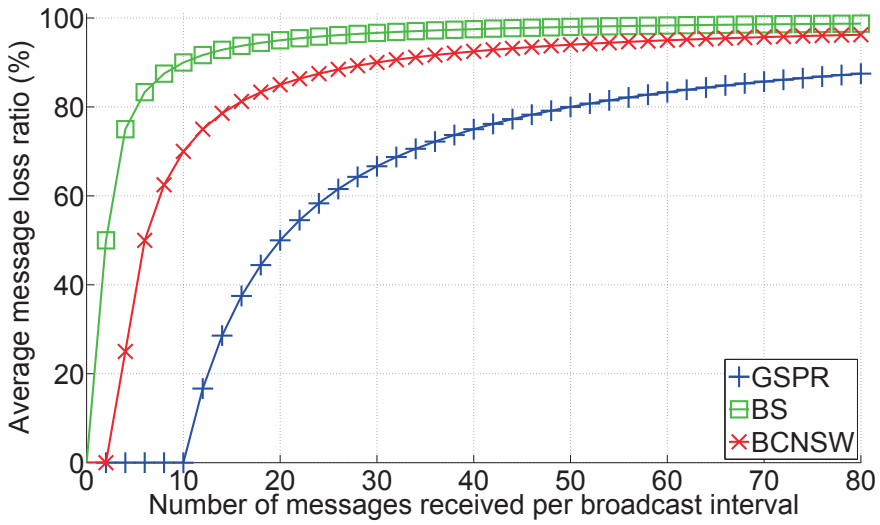


Figure 4.5: Average message loss ratio in GSs vs. number of messages received per broadcast interval.

4.6 Application

To illustrate the practical advantages of GSPR, in this section, we compare the signature verification performance of GSPR with two benchmarks (i.e., BS and BCNSW) for a specific type of applications, viz., vehicular network (VANET) safety applications. Since the allocation of the Dedicated Short-Range Communications (DSRC) spectrum in the 5.9 GHz band by the Federal Communications Commission (FCC), the automotive industry and the other stakeholders have been actively developing DSRC technologies, with a particular focus on vehicular safety applications.

In a typical safety application, each vehicle broadcasts beacon messages that contain information critical to safety, such as speed, direction of movement, acceleration, etc. The beacon messages need to be authenticated, but, at the same time, the privacy of the transmitting vehicle's driver must be protected [14]. Without such protection, adversaries can use the beacon messages to track the driver's movement or, worse yet, use them for more nefarious purposes. Hence, safety applications is one important application domain for

privacy-preserving authentication techniques.

Typically, beacon messages are broadcast in intervals of 100 ms [14]. In high vehicular density scenarios, a given vehicle is expected to receive a large number of beacon messages within a broadcast interval, and each message needs to be authenticated before the arrival of the next message from the same transmitter. If the authentication of the current message cannot be finished before the arrival of the next message, then the current message must be discarded because it is considered to contain "stale" information. To measure the impact of the computational cost of signature verification on the performance of safety applications, we employ the average message loss ratio, which is defined as the ratio between the number of beacon messages discarded due to signature verification latency and the total number of beacon messages received by a particular vehicle in a broadcast interval of 100 ms [14].

When simulating GSPR, we assume that each vehicle is on the road for 2 hours per day [18], and replaces its current alias token with a new one every minute, which equates to 120 alias tokens per day. The simulation results are shown in Figures 4.5 assuming $n_r = 64$. From this figure, we observe that GSPR's signature verification procedure is efficient enough to ensure acceptable performance for safety applications under reasonably-favorable conditions. In contrast, our results suggest that the computational burden of the verification procedures used by BS and BCNSW is too heavy for their use in vehicular safety applications.

4.7 Summary

In this chapter, we proposed a novel verifier-anonymous authentication scheme called Group Signatures with Probabilistic Revocation (GSPR). It is well known that revocation is the primary performance bottleneck of modern group signature schemes and that existing schemes do not scale well to large networks because of high computational cost of their revocation check procedures. By using the novel concept of probabilistic revocation, GSPR manages to significantly reduce the computational burden of the revocation check procedure at the

cost of increased communication overhead. The negative impact of the increased communication overhead can be mitigated by pre-fetching the revocation code from the issuer before signature verification.

Chapter 5

LASER: Lightweight Anonymous Attestation Scheme with Efficient Revocation

In this chapter, a novel DAA scheme called *Lightweight Anonymous attestation Scheme with Efficient Revocation* (LASER) is proposed. In LASER, the computational complexity and communication overhead of the signature generation and verification procedures are multiple orders of magnitude lower than the prior art. LASER achieves this significant performance improvement by shifting most of the computational complexity and communication overhead from a DAA's online procedure (i.e., signature generation and verification) to its offline procedure (i.e., acquisition of keys/credentials from the issuer). A comprehensive evaluation of LASER is conducted by implementing it on two computing platforms that are executing a TPM emulator. The security of LASER is thoroughly analyzed in the random oracle model under the $\mathbb{G}_1$-DL assumption, the $\mathbb{G}_1$-DDH assumption, and the q-SDH assumption discussed in Section 3.4.

The rest of this chapter is organized as follows. We present a brief overview of LASER in Section 5.1. We provide the model of LASER and the security definitions utilized for

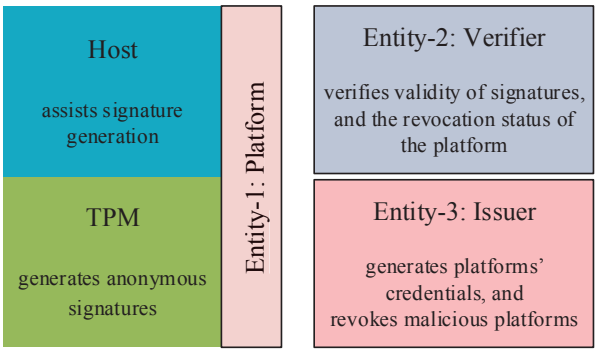


Figure 5.1: Entities in anonymous attestation scenario.

LASER in Section 5.2. We present the details of LASER in Section 5.3, and the the proofs of security in Section 5.4. We comprehensively analyze the computational and communication overheads of LASER in Section 5.5. We discuss the results obtained from the implementation of LASER in Section 5.6. We conclude the chapter in Section 5.7.

5.1 Overview of LASER

Here, we provide a brief overview of LASER. Figure 5.1 presents the functions performed by the three entities, the platform, the verifier and the issuer, in DAA. In LASER, firstly, the issuer sets up a group, and generate the group parameters. To join the group created by the issuer, the TPM generates a *TPM's secret key*. The platform runs a protocol with the issuer to obtain a *membership credential* corresponding to the TPM's secret key. Unlike the existing DAA scheme, in LASER, the platform does *not* utilize the same membership credential to also generate signatures which need to be sent to the verifier. Instead, in LASER, the platform, by utilizing the membership credential and the TPM's secret key, runs a protocol with the issuer to obtain a *signing credential*. The platform may run the protocol for multiple times to obtain multiple "unlinkable" signing credentials. In each signing credential, the platform also obtains multiple alias tokens. In this protocol, the platform also proves to the issuer that the TPM's secret key is not revoked by generating a

proof of knowledge for each tuple included in a basename-based revocation list.

Further, by utilizing the TPM's secret key, and a signing credential, the platform runs the signing protocol to sign a message. The platform reveals an alias token in the signature. The verifier runs the verification algorithm to verify the signed message. At the verifier, the revocation status of the platform is determined by checking the revocation status of the alias token. This is performed by searching through an alias token-based revocation list which contains all the revoked alias tokens.

To revoke a platform using its signature, a revocation algorithm is used by the issuer to revoke the alias token associated with the signature. Revoking an alias token of the platform is equivalent to revoking the platform's signing credential, all the corresponding alias tokens, the platform's membership credential, and the TPM's secret key. If the issuer obtains a TPM's secret key corresponding to a corrupted platform, it utilizes the revocation algorithm to directly revoke the TPM's secret key. Revoking the TPM's secret key is equivalent to revoking all the platform's alias tokens, all the signing credentials, and the membership credential. Here, we assume that the issuer also acts as the revocation manager.

Note that unlike the existing DAA schemes, in LASER, for the signatures sent to the verifier, the platform does not need to generate any proof of non-revocation of the TPM's secret key. This unique feature of LASER brings about a number of important practical advantages. First, during the signature generation procedure, the platform is *not* burdened with any computations related to the revocation check procedure, resulting in a significant reduction in the computation complexity of signature generation. Second, the signature length is constant, and does not grow proportionally with the length of the revocation list. Third, LASER enables the verifier to employ a computationally efficient procedure to check the revocation status of the platform that has issued a given signature by employing alias tokens.

5.2 Model and Security Definitions

Definition 5.1. *Lightweight Anonymous Attestation Scheme with Efficient Revocation*

LASER comprises of the following algorithms and protocols.

1. Setup: In this algorithm, with the security parameter 1^λ, where $\lambda \in \mathbb{N}$, as the input, the issuer generates a group public key `gpk`, and an issuer's secret key `isk`. Here, $\mathbb{N}$ represents the set of natural numbers. The group public key `gpk` is published.

2. GetMemCre: This protocol is performed among the TPM, the host and the issuer. Here, we assume that each TPM has a secret endorsement key, S_T, embedded into it, and there is an associated public key, P_T. We also assume that the issuer has a long-term public/secret key pair (P_I, S_I). These keys are utilized for authentication between the issuer and the TPM. Further, we assume that the TPM has an internal secret value, represented by `DAAseed`, and an internal counter value *cnt*. The counter value *cnt* is utilized to generate multiple TPM's secret keys for association with multiple groups. In this protocol, the TPM generates a TPM's secret key `tsk` and a corresponding public key `tpk`. The TPM's public key `tpk` is forwarded to the host. The TPM and the host generate a request to join the group created by the issuer, and send it to the issuer. In the response, the issuer, using its secret key `isk`, generates a set of parameters for a membership credential associated with the TPM's secret key `tsk`, and send it to the platform. After receiving the parameters, and verifying their validity, the host outputs the membership credential `memCre`. At the end of this protocol, the platform becomes a member of the group created by the issuer.

3. GetSignCre: This protocol is performed among the TPM, the host and the issuer. In this protocol, the TPM with `tsk` as the input, and the host with `tpk` and `memCre` as the inputs generate a signature σ_0 to prove the platform's membership of the group. For each $i \in [1, m_r]$, the TPM and the host also generate a signature σ_i to prove that

`tsk` is not used to generate the i^{th} tuple in a basename-based revocation list `baseRL`. Here, m_r represents the number of revoked signing credentials. The host sends the set of signatures $(\sigma_0, \sigma_1, \cdots \sigma_{m_r})$ to the issuer. After verifying the validity of the received signatures, the issuer, using its secret key `isk`, generates a set of m_a alias tokens, $x_{jk} \in \mathbb{Z}_p^*$, $\forall k \in [1, m_a]$, and corresponding alias keys. Here, $\mathbb{Z}_p^*$ represents the set of integers modulo p, and x_{jk} represents the k^{th} alias token of j^{th} signing credential. The issuer also makes entries of the alias tokens x_{jk}, $\forall k \in [1, m_a]$, into a database registry, `reg`. The issuer sends the set of alias tokens and corresponding alias keys to the platform. After verifying the validity of the received parameters, the host outputs the signing credential $\texttt{signCre}_\texttt{j}$ which comprises of the alias tokens and the alias keys.

4. Sign: This protocol is performed between the TPM and the host. The TPM with its secret key `tsk` as input, and the host with the signing credential `signCre` and a message M as inputs, output a signature σ_s associated with an alias token x_{jk}.

5. Verify: The inputs to this algorithm are the group public key `gpk`, a purported signature σ_s associated with an alias token x_{jk}, a message M, and an alias token-based revocation list `atRL`. This deterministic algorithm outputs the value ***valid*** if it verifies that σ_s is a valid signature on M, and the alias token x_{jk} embedded in σ_s has not been revoked; otherwise, it outputs the value `invalid`.

6. Revoke: This revocation algorithm comprises of two sub-algorithm RevokeSig and RevokeTPM, and only one of these sub-algorithms is performed based on the available inputs. The sub-algorithm RevokeSig is the signature-based revocation algorithm, and its inputs are the group public key `gpk`, the database registry `reg`, the basename-based revocation list `baseRL`, the alias token-based revocation list `atRL`, a message M and a purported signature σ_s associated with an alias token x_{jk}. Here, we assume that the signature σ_s has been proven to be malicious, and needs to revoked. The sub-algorithm RevokeTPM is the TPM's secret key-based revocation algorithm, and its inputs are the group public key `gpk`, the database registry `reg`, the basename-based

revocation list `baseRL`, the alias token-based revocation list `atRL`, a TPM's secret key `tsk` and a membership credential `memCre`. Here, we assume that the TPM's secret key `tsk` has been extracted from the TPM, and published. Both these sub-algorithms update the revocation lists, `atRL` and `baseRL`.

7. TokenOnlyLink: This deterministic algorithm takes the group public key `gpk`, two signatures σ_s and σ'_s with the corresponding alias tokens x_{jk} and $x_{j'k'}$, the corresponding messages M and M', and the corresponding alias token-based revocation lists `atRL` and `atRL`$'$ as inputs. It outputs the value ***valid*** if both the signatures are detected to be generated by the same platform; otherwise it outputs the value `invalid`.

8. TokenRegLink: This deterministic algorithm takes the database registry `reg`, the group public key `gpk`, two signatures σ_s and σ'_s with the corresponding alias tokens x_{jk} and $x_{j'k'}$, the corresponding messages M and M', and the corresponding alias token-based revocation lists `atRL` and `atRL`$'$ as inputs. It outputs the value ***valid*** if both the signatures are detected to be generated by the same platform; otherwise it outputs the value `invalid`.

9. Identify: This signature tracing algorithm takes the group public key `gpk`, a message M, a purported signature σ_s associated with an alias token x_{jk}, the alias token-based revocation list `atRL`, and a TPM's secret key `tsk` as inputs. This algorithm outputs the value ***valid*** if σ_s is proved to have been generated with the key `tsk`; otherwise it outputs the value `invalid`.

In the following discussion, we review the four security properties of LASER.

1. *Correctness*: This property requires that if a signature is generated by an honest platform, the signature verification algorithm and the signature tracing algorithm output the value ***valid***.

2. *User-controlled anonymity*: This notion requires the following two properties.

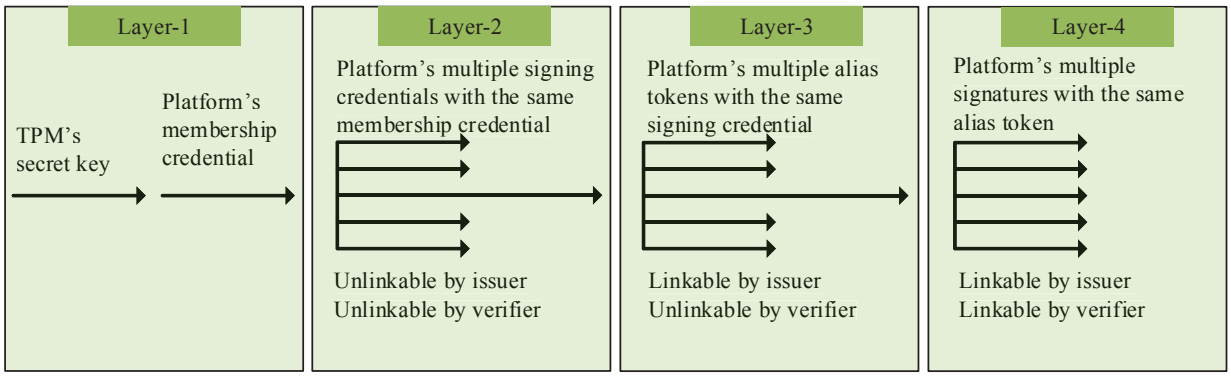


Figure 5.2: User-controlled unlinkablility in LASER.

(a) *Anonymity*: This property requires that no entity (including the issuer) is able to identify the platform which has generated a given signature.

(b) *User-controlled unlinkability*: This property requires that the platform is able to control whether or not two signatures, σ_s and σ'_s with corresponding alias tokens x_{jk} and $x_{j'k'}$, can be linked in the following way.

i. When $x_{jk} \neq x_{j'k'}$ and $j \neq j'$, no entity (including the issuer) is able to determine whether or not the two signatures are generated by the same platform.

ii. When $x_{jk} \neq x_{j'k'}$, $j = j'$, and $k \neq k'$, only the entity with the database registry `reg` as the input (i.e., the issuer) is able to determine whether or not the two signatures are generated by the same platform.

iii. When $x_{jk} = x_{j'k'}$, all entities (including the verifier) determine that the two signatures are generated by the same platform.

3. *Traceability*: This property requires that no colluding set of platforms (even consisting of the entire group) can create a valid signature that does not belong to any platform.

4. *Non-frameability*: This property requires that no colluding set of entities (including the issuer) can create a valid signature that belongs to any one of the non-colluding platforms.

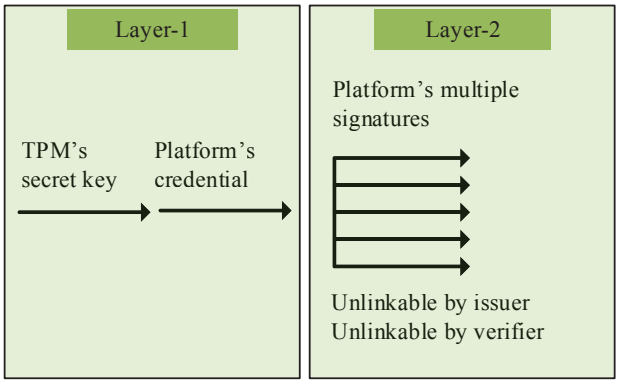


Figure 5.3: Unlinkablility in existing DAA schemes.

Among the above properties, the user-controlled unlinkability is an important property to consider in evaluating our proposed scheme, LASER, with respect to other DAA schemes. Figures 5.2 and 5.3 illustrate the unlinkability property in LASER and in the existing DAA schemes, respectively. In the existing DAA schemes, as shown in Figure 5.3, we observe that if the platform obtains a credential corresponding to the TPM's secret key (in Layer-1), all the signatures generated by the platform (in Layer-2) *can* be unlinkable by the verifier and the issuer. However, in LASER, as shown in Figure 5.2, we observe that the platform obtains a membership credential corresponding to the TPM's secret key in Layer-1, and multiple signing credentials, $\texttt{signCre}_j$, $\forall j \in [1, m_s]$, corresponding to the same membership credential in Layer-2. Recall that each signing credential $\texttt{signCre}_j$ contains multiple alias tokens x_{jk}, $\forall k \in [1, m_a]$, as shown in Layer-3. In Figure 5.2, we observe that if two signatures are generated by two different alias tokens corresponding to two different signing credentials, then they are unlinkable by the issuer as well as the verifier. If the two signatures are generated by two different alias tokens belonging to the same signing credential, then they are unlinkable by the verifier, but linkable by the issuer. But, if the same alias token is utilized in two different signature, the signatures can be easily linked by the verifier as well as the issuer. Hence, one of the intrinsic attributes of LASER that distinguishes it from all other DAA schemes is that it satisfies the user-controlled unlinkability property in a limited sense. LASER exploits this property to significantly reduce the large computational and

communication overheads plaguing the DAA scheme in the prior art.

Below, we provide formal definitions of the security properties mentioned above.

Definition 5.2. *Correctness*

For all $\lambda \in \mathbb{N}$, $M \in \{0,1\}^*$, $\mathtt{baseRL}$, $\mathtt{atRL}$, if

$$\begin{aligned}(\mathtt{gpk}, \mathtt{isk}) &\leftarrow \mathsf{Setup}(1^\lambda),\\ (\mathtt{tsk}, \mathtt{tpk}, \mathtt{memCre}) &\leftarrow \mathsf{GetMemCre}(\mathtt{gpk}, \mathtt{isk}),\\ \mathtt{signCre_j} &\leftarrow \mathsf{GetSignCre}(\mathtt{gpk}, \mathtt{isk}, \mathtt{tsk}, \mathtt{tpk},\\ &\qquad\qquad \mathtt{memCre}, \mathtt{baseRL}),\\ \sigma_s &\leftarrow \mathsf{Sign}(\mathtt{gpk}, \mathtt{tsk}, \mathtt{signCre}, M),\end{aligned}$$

then,

$$\begin{aligned}\mathit{valid} &\leftarrow \mathsf{Verify}(\mathtt{gpk}, \sigma_s, M, \mathtt{atRL})\\ \mathit{valid} &\leftarrow \mathsf{Identify}(\mathtt{gpk}, \sigma_s, M, \mathtt{atRL}, \mathtt{tsk}).\end{aligned}$$

Definition 5.3. *User-controlled anonymity*

For a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ and a challenger $\mathcal{C}$, the user controlled anonymity game is defined as follows.

1. *Setup*: The challenger $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda)$, and provides the adversary $\mathcal{A}$ with the resulting $\mathtt{isk}$ and $\mathtt{gpk}$.
2. *Queries-Phase I*: The adversary $\mathcal{A}$ can query the challenger $\mathcal{C}$ about the following.
   (a) *Sign*: The adversary $\mathcal{A}$ submits a TPM's identity ID, and a message M of its choice to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ runs Sign, and responds with the signature σ_s on M.

(b) *FetchMemCre*: The adversary $\mathcal{A}$ submits a new TPM's identity ID of its choice to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ runs $\mathsf{GetMemCre}$ with $\mathcal{A}$ to generate $\mathtt{memCre}$.

(c) *FetchSignCre*: The adversary $\mathcal{A}$ submits a TPM's identity ID of its choice to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ runs $\mathsf{GetSignCre}$, updates $\mathtt{reg}$, and responds with $\mathtt{signCre}_\mathtt{j}$.

(d) *FetchReg*: The adversary $\mathcal{A}$ requests the challenger $\mathcal{C}$ to provide the database registry $\mathtt{reg}$. The challnger $\mathcal{C}$ provides the current $\mathtt{reg}$ to the adversary $\mathcal{A}$.

(e) *Corrupt*: The adversary $\mathcal{A}$ submits a TPM's identity ID of its choice to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ responds with the TPM's secret key $\mathtt{tsk}$.

3. *Challenge*: The adversary $\mathcal{A}$ outputs a message M and two TPM's identities ID_0 and ID_1 with the restriction that the *Corrupt* queries for ID_0 and ID_1 have not been requested. The challenger $\mathcal{C}$ selects $\phi \xleftarrow{R} \{0, 1\}$, and responds with the signature on M for TPM ID_ϕ. Here, $\xleftarrow{R}$ represents a random selection.

4. *Queries-Phase II (Restricted Queries)*: After obtaining the challenge, the adversary $\mathcal{A}$ continues to probe the challenger $\mathcal{C}$ with the same types of queries that were discussed in *Queries-Phase I*, except for the corruption queries of ID_0 and ID_1.

5. *Output*: The adversary $\mathcal{A}$ outputs a bit ϕ' indicating its guess of ϕ.

The adversary $\mathcal{A}$ wins the above game if $\phi' = \phi$. The advantage of $\mathcal{A}$ is defined as $|\Pr(\phi' = \phi) - 1/2|$. LASER is user-controlled anonymous if for any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ on winning the above user-controlled anonymity game is negligibly small.

Definition 5.4. *Traceability*

For a PPT adversary $\mathcal{A}$ and a challenger $\mathcal{C}$, the traceability game is defined as follows.

1. *Setup*: The challenger $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda)$, and provides the adversary $\mathcal{A}$ with the resulting $\mathtt{gpk}$. The challenger $\mathcal{C}$ does not reveal $\mathtt{isk}$ to the adversary $\mathcal{A}$.

2. *Queries*: The adversary $\mathcal{A}$ queries the challenger $\mathcal{C}$ about the following.
   (a) *Sign*: The same as the *Sign* query in the user-controlled anonymity game.
   (b) *FetchMemCre*: The same as *FetchMemCre* query in the user-controlled anonymity game.
   (c) *FetchSignCre*: The same as *FetchSignCre* query in the user-controlled anonymity game.
   (d) *Corrupt*: The same as *Corrupt* query in the user-controlled anonymity game.
3. *Output*: The adversary $\mathcal{A}$ outputs a message M^* and a signature σ_s^* associated with an alias token x_{jk}^*.

The adversary $\mathcal{A}$ wins the game if:

1. $\mathsf{Verify}(\mathtt{gpk}, \sigma_s^*, M^*, \mathtt{atRL}) = \mathtt{valid}$;
2. The adversary $\mathcal{A}$ did not obtain σ_s^* by making a *Sign* query;
3. For any TPM with its secret key $\mathtt{tsk}$, $\mathsf{Identify}(\mathtt{gpk}, \sigma_s^*, M^*, \mathtt{atRL}, \mathtt{tsk}) = \mathtt{invalid}$.
4. For any signature σ_s' honestly generated by a platform on any message M', $\mathsf{TokenRegLink}(\mathtt{reg}, \mathtt{gpk}, \sigma_s^*,$ $\mathtt{invalid}$. This means that $x^* \notin \mathtt{reg}$.

LASER is traceable if for any PPT algorithm $\mathcal{A}$, the probability that $\mathcal{A}$ wins the above traceability game is negligibly small.

Definition 5.5. *Non-frameability*

For a PPT adversary $\mathcal{A}$ and a challenger $\mathcal{C}$, the non-framability game is defined as follows.

1. *Setup*: The challenger $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda)$, and provides the adversary $\mathcal{A}$ with the resulting $\mathtt{isk}$ and $\mathtt{gpk}$. Also, the challenger $\mathcal{C}$ sets the array of the identities of the corrupted TPMs, $\mathtt{U}$, as empty.

2. *Queries*: The adversary $\mathcal{A}$ queries the challenger about the following.
   (a) *Sign*: The same as the *Sign* query in the user-controlled anonymity game.
   (b) *FetchMemCre*: The same as *FetchMemCre* query in the user-controlled anonymity game.
   (c) *FetchSignCre*: The same as *FetchSignCre* query in the user-controlled anonymity game.
   (d) *Corrupt*: The adversary $\mathcal{A}$ submits a TPM's identity ID of its choice to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ responds with the TPM's secret key $\mathtt{tsk}$. Afterwards, the challenger $\mathcal{C}$ appends ID to $\mathtt{U}$.
3. *Output*: The adversary $\mathcal{A}$ outputs a TPM's identity ID^*, a message M^* and a signature σ_s^*.

The adversary $\mathcal{A}$ wins the game if:

1. $ID^* \notin \mathtt{U}$;
2. $\mathsf{Verify}(\mathtt{gpk}, \sigma_s^*, M^*, \mathtt{atRL}) = \mathit{valid}$;
3. The adversary $\mathcal{A}$ did not obtain σ^* by making a *Sign* query;
4. $\mathsf{Identify}(\mathtt{gpk}, \sigma_s^*, M^*, \mathtt{atRL}, \mathtt{tsk}) = \mathit{valid}$, where $\mathtt{tsk}$ is the secret key of the TPM with identity ID^*.

LASER is non-frameable if for any PPT adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins the above non-framability game is negligibly small.

5.3 Details of LASER

In the following subsections, we present the details of the algorithms and the protocols that make up LASER.

5.3.1 Setup

With the security parameter 1^λ as the input, the issuer runs this algorithm to output an issuer's secret key `isk`, and a corresponding group public key `gpk`. This algorithm performs the following.

1. Select an asymmetric bilinear group pair ($\mathbb{G}_1$,$\mathbb{G}_2$) of prime order p. Let g_1 and g_2 be the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively.
2. Select a bilinear mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. We provide the discussion of the bilinear mapping e in Section 3.1.
3. Select two hash functions $H_z : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, and $H_g : \mathbb{Z}_p^* \rightarrow \mathbb{G}_1$. Here, H_z and H_g are assumed to be collision resistant, and are treated as random oracles. We provide the discussion of the implementation of H_z and H_g in Section 3.2.
4. Select $h_1, h_2, h_3 \xleftarrow{R} \mathbb{G}_1$.
5. Select $\gamma \xleftarrow{R} \mathbb{Z}_p^*$, and compute $\omega = g_2^\gamma$.
6. Output the issuer's secret key, $\texttt{isk} = \gamma$, and the group public key, $\texttt{gpk} = (p, \mathbb{G}_1, \mathbb{G}_2, e, H_z, H_g, g_1, h_1, h_2, h_3, g_2, \omega)$.
7. Publish the group public key, `gpk`.

5.3.2 GetMemCre

This protocol is performed among the TPM, the host and the issuer. The inputs to the TPM are the group public key `gpk`, the internal counter value *cnt*, and the embedded secret key `DAAseed`. The inputs to the host are the issuer's long term public key P_I, and the group public key `gpk`. The inputs to the issuer are the group public key `gpk`, and the issuer's secret key `isk`. In this protocol, the TPM outputs the TPM's secret key `tsk` and the TPM's public

key `tpk`, and the host outputs the platform's membership credential `memCre`. The protocol proceeds as follows.

1. The host sends a membership credential request to the issuer.
2. The issuer selects a nonce $n_{im} \xleftarrow{R} \{0,1\}^{\lambda}$, generates a record corresponding to n_{im}, and sends n_{im} to the host.
3. The host and the TPM performs the following to generate the TPM's public/secret key pair.
   (a) The host forwards the issuer's public key P_I to the TPM.
   (b) The platforms performs the following.
      i. Compute $f = H_z(\texttt{DAAseed} \parallel P_I \parallel cnt)$. Here, $\parallel$ represents concatenation of two strings of bits.
      ii. Compute $I = h_1^f$.
      iii. Output the secret key belonging only to the TPM, $\texttt{tsk} = f$, and the TPM's public key, $\texttt{tpk} = I$.
      iv. Forward I to the host.
4. Further, the host and the TPM perform the following steps to generate a signature σ_m for the platform's membership credential request.
   (a) The TPM performs the following steps.
      i. Select $r_{fm} \xleftarrow{R} \mathbb{Z}_p^*$, and compute $R_m = h_1^{r_{fm}}$.
      ii. Forward R_m to the host.
   (b) The host computes $c_{hm} = H_z(\texttt{gpk} \parallel n_{im} \parallel I \parallel R_m)$, and forwards c_{hm} to the TPM.
   (c) The TPM performs the following steps.
      i. Select a nonce $n_{tm} \xleftarrow{R} \{0,1\}^{\lambda}$, and compute $c_{tm} = H_z(c_{hm} \parallel n_{tm})$.

ii. Compute $s_{fm} = r_{fm} + c_{tm} \cdot f$.

iii. Forward (n_{tm}, c_{tm}, s_{fm}) to the host.

(d) The host performs the following

i. Output the signature $\sigma_m = (I, n_{tm}, c_{tm}, s_{fm})$.

ii. Send (n_{im}, σ_m) to the issuer.

Note that in this protocol, the signature σ_m presents the proof of knowledge given by

$$PK\{f : I = h_1^f\}.$$

5. For the record corresponding to n_{im}, the issuer verifies that the signature σ_m is valid as follows.

(a) Retrieve $\widetilde{R}_m = h_1^{s_{fm}} \cdot I^{-c_{tm}}$.

(b) Compute $\widetilde{c}_{hm} = H_z(\texttt{gpk} \parallel n_{im} \parallel I \parallel \widetilde{R}_m)$, and verify that $c_{tm} \stackrel{?}{=} H_z(\widetilde{c}_{hm} \parallel n_{tm})$. If the verification fails, abort.

6. The issuer proceeds to generate the parameters for the platform's membership credential as follows.

(a) Select $z, \rho \xleftarrow{R} \mathbb{Z}_p^*$, and compute $J = (g_1 \cdot I \cdot h_2^{\rho})^{\frac{1}{\gamma+z}}$.

(b) Send (J, z, ρ) to the host.

7. The host verifies the parameters for the platform's membership credential, and accepts them if the verification succeeds as follows.

(a) Verify that $e(J, \omega \cdot g_2^z) = e(g_1 \cdot I \cdot h_2^{\rho}, g_2)$. If the verification fails, abort.

(b) Output the platform's membership credential, $\texttt{memCre} = (J, z, \rho)$.

5.3.3 GetSignCre

This protocol is performed among the TPM, the host and the issuer. The inputs to the TPM are the group public key `gpk`, and the TPM's secret key `tsk`. The inputs to the host are the group public key `gpk`, the platform's membership credential `memCre`, the TPM's public key `tpk`, and a basename-based revocation list `baseRL`. The inputs to the issuer are the group public key `gpk`, the issuer's secret key `isk`, the basename-based revocation list `baseRL`, and a database registry `reg`. Here, the basename-based revocation list is represented as

$$\texttt{baseRL} = \{(a_i, b_{2i}, K_i) : a_i, b_{2i} \in \mathbb{Z}_p^*, K_i \in \mathbb{G}_1, \forall i \in [1, m_r]\}.$$

The output of the protocol is a signing credential for the platform containing m_a alias tokens which can subsequently be used for generating signatures on the messages. In essence, in this protocol, the host and the TPM generate a signing credential request with signatures on nonces to prove the platform's membership of the group to the issuer. After authenticating the membership, the issuer proceeds to generate the platform's signing credential. The protocol proceeds as follows.

1. The host and the TPM perform the following steps to generate a signature σ_0 to prove the platform's membership of the group.
   (a) The host performs the following.
      i. Select $a_j \xleftarrow{R} \mathbb{Z}_p^*$, and compute the basename $B_j = (b_{1j}, b_{2j}) = H_g(a_j)$.
      ii. Forward (a_j, b_{2j}) to the TPM.
   (b) The TPM performs the following.
      i. Compute $b_{1j} = H_z(a_j)$, and set $B_j = (b_{1j}, b_{2j})$.
      ii. Compute $K_j = B_j^f$.
      iii. Select $r_{fg} \xleftarrow{R} \mathbb{Z}_p^*$, and compute $S_{10} = B_j^{r_{fg}}$, and $S_{20} = h_1^{r_{fg}}$.
      iv. Forward (K_j, S_{10}, S_{20}) to the host.

(c) The host performs the following.

i. Select $\alpha, \xi \xleftarrow{R} \mathbb{Z}_p^*$; and compute $\eta = \xi \cdot z$, $L_1 = I \cdot h_2^{\alpha}$, and $L_2 = J \cdot h_3^{\xi}$.

ii. Select $r_\alpha, r_\rho, r_z, r_\xi, r_\eta \xleftarrow{R} \mathbb{Z}_p^*$, and compute

$$\begin{aligned} R_{10} &= S_{10}, \\ R_{20} &= S_{20} \cdot h_2^{r_\alpha}, \\ R_{30} &= e(L_2, g_2)^{-r_z} \cdot e(S_{20}, g_2) \cdot e(h_2, g_2)^{r_\rho} \cdot e(h_3, \omega)^{r_\xi} \cdot e(h_3, g_2)^{r_\eta}, \\ &= e(L_2^{-r_z} \cdot S_{20} \cdot h_2^{r_\rho} \cdot h_3^{r_\eta}, g_2) \cdot e(h_3, \omega)^{r_\xi}. \end{aligned}$$

iii. Compute $c_{h0} = H_z(\texttt{gpk} \parallel B_j \parallel K_j \parallel L_1 \parallel L_2 \parallel R_{10} \parallel R_{20} \parallel R_{30})$.

iv. Forward c_{h0} to the TPM.

(d) The TPM performs the following.

i. Select a nonce $n_{t0} \xleftarrow{R} \{0,1\}^{\lambda}$, and compute $c_{t0} = H_z(c_{h0} \parallel n_{t0})$.

ii. Compute $s_{fg} = r_{fg} + c_{t0} \cdot f$.

iii. Forward (n_{t0}, c_{t0}, s_{fg}) to the host.

(e) The host performs the following.

i. Compute $s_\alpha = r_\alpha + c_{t0} \cdot \alpha$, $s_\rho = r_\rho + c_{t0} \cdot \rho$, $s_z = r_z + c_{t0} \cdot z$, $s_\xi = r_\xi + c_{t0} \cdot \xi$, and $s_\eta = r_\eta + c_{t0} \cdot \eta$.

ii. Output the signature $\sigma_0 = (a_j, b_{2j}, K_j, L_1, L_2, n_{t0}, c_{t0}, s_{fg}, s_\alpha, s_\rho, s_z, s_\xi, s_\eta)$.

Note that in this protocol, the signature σ_0 presents the proof of knowledge given by

$$PK\{(f, \alpha, z, \rho, \xi) : K_j = B_j^f,\ L_1 = I \cdot h_2^{\alpha},\ L_2 = J \cdot h_3^{\xi},\ e(J, \omega \cdot g_2^z) = e(g_1 \cdot I \cdot h_2^{\rho}, g_2)\}.$$

which can also be represented as

$$\begin{aligned} PK\{(f, \alpha, z, \rho, \xi) : K_j = B_j^f,\ L_1 = h_1^f \cdot h_2^{\alpha}, \\ e(L_2, \omega) \cdot e(g_1, g_2)^{-1} = e(L_2, g_2)^{-z} \cdot e(h_1, g_2)^f \cdot e(h_2, g_2)^{\rho} \cdot e(h_3, \omega)^{\xi} \cdot e(h_3, g_2)^{\xi \cdot z}\}. \end{aligned}$$

2. Further, for each $i \in [1, m_r]$, the host and the TPM perform the following steps to generate a signature σ_i. The honestly generated signatures $\sigma_i, \forall i \in [1, m_r]$, act as the proof that the TPM's secret key `tsk` is not revoked.

   (a) The host does the following.

      i. Compute $b_{1i} = H_z(a_i)$.

      ii. Set the basename, $B_i = (b_{1i}, b_{2i})$, and forward (a_i, b_{2i}, B_j) to the TPM.

   (b) The TPM performs the following.

      i. Compute $b_{1i} = H_z(a_i)$, and set $B_i = (b_{1i}, b_{2i})$.

      ii. Compute $O_i = B_i^f$.

      iii. Select $r_{fi} \xleftarrow{R} \mathbb{Z}_p^*$; and compute $S_{1i} = B_i^{r_{fi}}$, and $S_{2i} = B_j^{r_{fi}}$.

      iv. Forward (O_i, S_{1i}, S_{2i}) to the host.

   (c) The host performs the following.

      i. Verify that $K_i \neq O_i$. If the verification fails, abort. Note that $K_i = O_i$, if $K_i = B_i^f$.

      ii. Select $\tau_i \xleftarrow{R} \mathbb{Z}_p^*$; and compute $P_i = (O_i \cdot K_i^{-1})^{\tau_i}$.

      iii. Select $r_{\tau i} \xleftarrow{R} \mathbb{Z}_p^*$; and compute

$$R_{1i} = S_{1i}^{\tau_i} \cdot K_i^{-r_{\tau i}},$$
$$R_{2i} = S_{2i}^{\tau_i} \cdot K_j^{-r_{\tau i}}.$$

      iv. Compute $c_{hi} = H_z(B_j \parallel K_j \parallel B_i \parallel K_i \parallel P_i \parallel R_{1i} \parallel R_{2i})$.

      v. Forward c_{hi} to the TPM.

   (d) The TPM performs the following.

      i. Select a nonce $n_{ti} \xleftarrow{R} \{0,1\}^\lambda$, and compute $c_{ti} = H_z(c_{hi} \parallel n_{ti})$.

      ii. Compute $s_{fi} = r_{fi} + c_{ti} \cdot f$.

      iii. Forward (n_{ti}, c_{ti}, s_{fi}) to the host.

(e) The host performs the following.

i. Compute $s_{\tau i} = r_{\tau i} + c_{ti} \cdot \tau_i$, and $s_{\nu i} = \tau_i \cdot s_{fi}$.

ii. Output the signature $\sigma_i = (P_i, n_{ti}, c_{ti}, s_{\tau i}, s_{\nu i})$.

Note that in this protocol, the signature σ_i presents the proof of knowledge given by

$$PK\{(f, \tau_i) : \nu_i = \tau_i \cdot f,\ P_i = B_i^{\nu_i} \cdot K_i^{-\tau_i},\ K_j = B_j^f\},$$

which can also be represented as

$$PK\{(f, \tau_i) : P_i = B_i^{\nu_i} \cdot K_i^{-\tau_i}, 1 = B_j^{\nu_i} \cdot K_j^{-\tau_i}\}.$$

3. The host sends $(\sigma_0, \sigma_1, \cdots, \sigma_{m_r})$ to the issuer.

4. The issuer verifies the validity of signature σ_0 to ascertain the platform's membership of the group as follows.

(a) Compute $b_{1j} = H_z(a_j)$, and set $B_j = (b_{1j}, b_{2j})$

(b) Retrieve

$$\begin{aligned}
\widetilde{R}_{10} =&\ B_j^{s_{f0}} \cdot K_j^{-c_{t0}}, \\
\widetilde{R}_{20} =&\ h_1^{s_{f0}} \cdot h_2^{s_\alpha} \cdot L_1^{-c_{t0}}, \\
\widetilde{R}_{30} =&\ e(L_2, g_2)^{-s_z} \cdot e(h_1, g_2)^{s_{f0}} \cdot e(h_2, g_2)^{s_\rho} \cdot e(h_3, \omega)^{s_\xi} \cdot e(h_3, g_2)^{s_\eta} \cdot \\
&\ e(L_2, \omega)^{-c_{t0}} \cdot e(g_1, g_2)^{c_{t0}} \\
=&\ e(L_2^{-s_z} \cdot h_1^{s_{f0}} \cdot h_2^{s_\rho} \cdot h_3^{s_\eta} \cdot g_1^{c_{t0}}, g_2) \cdot e(h_3^{s_\xi} \cdot L_2^{-c_{t0}}, \omega).
\end{aligned}$$

(c) Compute $\widetilde{c}_{h0} = H_z(\texttt{gpk} \parallel B_j \parallel K_j \parallel L_1 \parallel L_2 \parallel \widetilde{R}_{10} \parallel \widetilde{R}_{20} \parallel \widetilde{R}_{30})$, and verify that $c_{t0} \stackrel{?}{=} H_z(\widetilde{c}_{h0} \parallel n_{t0})$. If the verification fails, abort.

5. For each $i \in [1, m_r]$, the issuer checks the revocation status of the TPM's secret key by verifying the validity of signature σ_i using the following steps.

(a) Compute $b_{1i} = H_z(a_i)$, and set $B_i = (b_{1i}, b_{2i})$

(b) Retrieve

$$\widetilde{R}_{1i} = B_i^{s_{\nu i}} \cdot K_i^{-s_{\tau i}} \cdot P_i^{-c_{ti}},$$
$$\widetilde{R}_{2i} = B_j^{s_{\nu i}} \cdot K_j^{-s_{\tau i}}.$$

(c) Compute $\widetilde{c}_{hi} = H_z(B_j \parallel K_j \parallel B_i \parallel K_i \parallel P_i \parallel \widetilde{R}_{1i} \parallel \widetilde{R}_{2i})$, and verify that $c_{ti} \stackrel{?}{=} H_z(\widetilde{c}_{hi} \parallel n_{ti})$. If the verification fails, abort.

(d) Verify that $P_i \neq 1$. If the verification fails, abort.

6. The issuer proceeds to generate the parameters of the platform's signing credential as follows.

(a) Select $\beta \xleftarrow{R} \mathbb{Z}_p^*$.

(b) For each $k \in [1, m_a]$, select an alias token, $x_{jk} \xleftarrow{R} \mathbb{Z}_p^*$, and compute $A_{jk} = (g_1 \cdot L_1 \cdot h_2^{\beta})^{\frac{1}{\gamma + x_{jk}}}$.

(c) Append an entry of the tuple $(a_j, b_{2j}, K_j, x_{j1}, \cdots, x_{jm_a})$ to the database registry `reg`.

(d) Send $(\beta, A_{j1}, x_{j1}, \cdots, A_{jm_a}, x_{jm_a})$ to the host.

7. The host verifies the validity of the received parameters for the platform's signing credential as follows.

(a) Compute $y_j = \alpha + \beta$.

(b) Verify that $e(A_{jk}, \omega \cdot g_2^{x_{jk}}) = e(g_1 \cdot L_1 \cdot h_2^{\beta}, g_2)$, $\forall k \in [1, m_a]$. If the verification fails, abort.

(c) Output the signing credential as

$$\mathtt{signCre_j} = (y_j, A_{j1}, x_{j1}, \cdots, A_{jm_a}, x_{jm_a}).$$

5.3.4 Sign

This protocol is performed by the TPM and the host. The inputs to the TPM are the group public key `gpk`, and the TPM's secret key `tsk`. The inputs to the host are the group public key `gpk`, the signing credentials `signCre`, and a message to be signed $M \in \{0,1\}^*$. This protocol outputs a signature σ_s by the following steps.

1. The host initiates the generation of the signature on message M as follows.
   (a) Select a tuple (y_j, A_{jk}, x_{jk}) from `signCre` by selecting some value of $j \in [1, m_s]$ and $k \in [1, m_a]$. In this paper, we do not discuss the specific details about this selection. Here, we assume that the host employs an algorithm for this selection considering the user-controlled unlinkability property discussed in Section 5.2.
   (b) Select $t \xleftarrow{R} \mathbb{Z}_p^*$, and compute $D = (d_1, d_2) = H_g(t)$.
   (c) Forward (t, d_2) to the TPM.
2. The TPM does the following.
   (a) Compute $d_1 = H_z(t)$, and set $D = (d_1, d_2)$.
   (b) Compute $E = D^f$.
   (c) Select $r_{fs} \xleftarrow{R} \mathbb{Z}_p^*$, and compute $S_{1s} = D^{r_{fs}}$, and $S_{2s} = h_1^{r_{fs}}$.
   (d) Forward (E, S_{1s}, S_{2s}) to the host.
3. The host does the following.
   (a) Select $\delta \xleftarrow{R} \mathbb{Z}_p^*$; and compute $T = A_{jk} \cdot h_3^{\delta}$.
   (b) Select $r_y, r_\delta \xleftarrow{R} \mathbb{Z}_p^*$, and compute

$$\begin{aligned} R_{1s} &= S_{1s}, \\ R_{2s} &= e(S_{2s}, g_2) \cdot e(h_2, g_2)^{r_y} \cdot e(h_3, \omega \cdot g_2^{x_{jk}})^{r_\delta} \\ &= e(S_{2s} \cdot h_2^{r_y} \cdot h_3^{x_{jk} \cdot r_\delta}, g_2) \cdot e(h_3, \omega)^{r_\delta}. \end{aligned}$$

(c) Compute $c_{hs} = H_z(\texttt{gpk} \parallel D \parallel E \parallel T \parallel R_{1s} \parallel R_{2s})$.

(d) Forward (c_{hs}, M) to the TPM.

4. The TPM does the following.

(a) Select $n_{ts} \xleftarrow{R} \{0,1\}^\lambda$, and compute $c_{ts} = H_z(c_{hs} \parallel n_{ts} \parallel M)$.

(b) Compute $s_{fs} = r_{fs} + c_{ts} \cdot f$.

(c) Forward (n_{ts}, c_{ts}, s_{fs}) to the host.

5. The host does the following.

(a) Compute $s_y = r_y + c_{ts} \cdot y_j$, and $s_\delta = r_\delta + c_{ts} \cdot \delta$.

(b) Output the signature $\sigma_s = (x_{jk}, t, d_2, E, T, n_{ts}, c_{ts}, s_{fs}, s_y, s_\delta)$.

(c) Send (σ_s, M) to the verifier.

Note that in this protocol, the signature σ_s presents the proof of knowledge given by

$$SPK\{(f, y_j, \delta) : E = D^f,\ T = A_{jk} \cdot h_3^\delta,\ e(A_{jk}, \omega \cdot g_2^{x_{jk}}) = e(g_1 \cdot h_1^f \cdot h_2^{y_j}, g_2)\}(M),$$

which can also be represented as

$$SPK\{(f, y_j, \delta) : E = D^f,\ e(T, \omega \cdot g_2^{x_{jk}}) \cdot e(g_1, g_2)^{-1} = e(h_1, g_2)^f \cdot e(h_2, g_2)^{y_j} \cdot e(h_3, \omega \cdot g_2^{x_{jk}})^\delta\}(M).$$

5.3.5 Verify

This verification algorithm takes the group public key gpk, a message M, a purported signature σ_s, and an alias token-based revocation list atRL as inputs. Here, the alias token-based revocation list is represented as

$$\texttt{atRL} = \{x_{il} : x_{il} \in \mathbb{Z}_p^*, \forall i \in [1, m_r], \forall l \in [1, m_a]\}.$$

This algorithm verifies: (1) whether the signature was honestly generated, and (2) the revocation status of the alias token used to generate the signature. If both of these verification steps (as shown below) output ***valid***, this algorithm outputs ***valid***; otherwise outputs `invalid`.

1. Verify the validity of the signature σ_s as follows.

   (a) Compute $d_1 = H_z(t)$, and set $D = (d_1, d_2)$.

   (b) Retrieve

   $$\begin{aligned}\widetilde{R}_{1s} &= D^{s_{fs}} \cdot E^{-c_{ts}},\\ \widetilde{R}_{2s} &= e(h_1, g_2)^{s_{fs}} \cdot e(h_2, g_2)^{s_y} \cdot e(h_3, \omega \cdot g_2^{x_{jk}})^{s_\delta} \cdot e(T, \omega \cdot g_2^{x_{jk}})^{-c_{ts}} \cdot e(g_1, g_2)^{c_{ts}}\\ &= e(h_1, g_2)^{s_{fs}} \cdot e(h_2, g_2)^{s_y} \cdot e(g_1, g_2)^{c_{ts}} \cdot e(T^{-c_{ts}} \cdot h_3^{s_\delta}, \omega \cdot g_2^{x_{jk}}).\end{aligned}$$

   (c) Compute $\widetilde{c}_{hs} = H_z(\texttt{gpk} \parallel D \parallel E \parallel T \parallel \widetilde{R}_{1s} \parallel \widetilde{R}_{2s})$, and verify that $c_{ts} \stackrel{?}{=} H_z(\widetilde{c}_{hs} \parallel n_{ts} \parallel M)$. If the verification is successful, output ***valid***; otherwise output `invalid`.

2. Verify that $x_{jk} \notin \texttt{atRL}$ by utilizing a conventional binary search algorithm. If the verification is successful, output ***valid***; otherwise output `invalid`.

5.3.6 Revoke

This revocation algorithm comprises of two sub-algorithms, and only one of these sub-algorithms is performed based on the available inputs.

1. RevokeSig: This is the signature based revocation algorithm. The inputs to this sub-algorithm are the group public key `gpk`, the database registry `reg`, the basename-based revocation list `baseRL`, the alias token-based revocation list `atRL`, a message M and a purported signature σ_s associated with an alias token x_{jk}. This sub-algorithm updates the revocation lists, `atRL` and `baseRL`, using the following steps.

(a) Verify that σ_s is an honest signature, and the alias token x_{jk} embedded in the signature σ_s has not been revoked previously, i.e., $\mathsf{Verify}(\mathtt{gpk}, \sigma_s, M, \mathtt{atRL}) = \textit{valid}$. If the verification fails, abort.

(b) Search in the database registry `reg` for the tuple $(a_i, b_{2i}, K_i, x_{i1}, \cdots, x_{im_a})$ which contains x_{jk}.

(c) Append all the corresponding alias tokens, $(x_{i1}, \cdots, x_{im_a})$, to the alias token-based revocation list `atRL`. Note that this revokes the signing credential, $\mathtt{signCre_j}$.

(d) Append the corresponding (a_i, b_{2i}, K_i) to the basename-based revocation list `baseRL`. Note that this revokes the membership credential `memCre`, and the TPM's secret key `tsk`.

2. RevokeTPM: This is the TPM's secret key based revocation algorithm. The inputs to this sub-algorithm are the group public key `gpk`, the database registry `reg`, the basename-based revocation list `baseRL`, the alias token-based revocation list `atRL`, a TPM's secret key `tsk` and a membership credential `memCre`. This sub-algorithm updates the revocation lists, `atRL` and `baseRL`, using the following steps.

(a) Verify that `memCre` is a valid membership credential, i.e., $e(J, \omega \cdot g_2^z) = e(g_1 \cdot h_1^f \cdot h_2^\rho, g_2)$. If the verification fails, abort.

(b) Verify that `memCre` has not been revoked previously, i.e., for each tuple (a_i, b_{2i}, K_i) in `baseRL`, where $i \in [1, m_r]$, compute $b_{1i} = H_z(a_i)$, set $B_i = (b_{1i}, b_{2i})$, and verify that $K_i \neq B_i^f$. If the verification fails, abort.

(c) Search in the database registry `reg` for all the tuples $(a_l, b_{2l}, K_l, x_{l1}, \cdots, x_{lm_a})$ for which $K_l = B_l^f$, where $b_{1l} = H_z(a_l)$, and $B_l = (b_{1l}, b_{2l})$.

(d) Append all the corresponding alias tokens, $(x_{l1}, \cdots, x_{lm_a})$, to the alias token-based revocation list `atRL`. Note that this revokes all the signing credentials `signCre`.

(e) Append all the corresponding (a_l, b_{2l}, K_l) to the basename-based revocation list `baseRL`. Note that this revokes the membership credential `memCre`, and the TPM's

secret key `tpk`.

5.3.7 TokenOnlyLink

The inputs to this algorithm are the group public key `gpk`, two signatures σ_s and σ'_s corresponding to alias tokens x_{jk} and $x'_{j'k'}$, corresponding messages M and M', and corresponding alias token-based revocation lists `atRL` and `atRL'`. This algorithm outputs ***valid*** if the verifier links the two signatures, σ_s and σ'_s to the same platform; otherwise it outputs `invalid`.

1. Verify that σ_s and σ'_s are honest signatures, i.e., $\mathsf{Verify}(\mathtt{gpk}, \sigma_s, M, \mathtt{atRL}) =$ ***valid***, and $\mathsf{Verify}(\mathtt{gpk}, \sigma'_s, M', \mathtt{atRL}') =$ ***valid***.
2. If $x_{jk} = x'_{j'k'}$, output ***valid***; otherwise output `invalid`.

5.3.8 TokenRegLink

The inputs to this algorithm are the database registry `reg`, the group public key `gpk`, two signatures σ_s and σ'_s with corresponding alias tokens x_{jk} and $x'_{j'k'}$, corresponding messages M and M', and corresponding alias token-based revocation lists `atRL` and `atRL'`. This algorithm outputs ***valid*** if the issuer links the two signatures σ_s and σ'_s to the same platform; otherwise it outputs `invalid`.

1. Verify that σ_s and σ'_s are honest signatures, i.e., $\mathsf{Verify}(\mathtt{gpk}, \sigma_s, M, \mathtt{atRL}) =$ ***valid***, and $\mathsf{Verify}(\mathtt{gpk}, \sigma'_s, M', \mathtt{atRL}') =$ ***valid***.
2. Search the database registry `reg` for the tuple $(a_i, b_{2i}, K_i, x_{i1}, \cdots, x_{im_a})$ which contains x_{jk}.
3. In the tuple $(x_{i1}, \cdots, x_{im_a})$, if $x'_{j'k'} = x_{il}$ for any $l \in [1, m_a]$, output ***valid***; otherwise output `invalid`.

5.3.9 Identify

This algorithm takes the group public key `gpk`, a message M, a purported signature σ_s associated with an alias token x_{jk}, the alias token-based revocation list `atRL`, and a TPM's secret key `tsk` as inputs. This algorithm outputs the value ***valid*** if the following two steps succeed (i.e., σ_s is proved to have been generated with the key `tsk`); otherwise it outputs the value `invalid`.

1. Verify that σ_s is an honest signature, i.e., $\mathsf{Verify}(\mathtt{gpk}, \sigma_s, M, \mathtt{atRL}) = \textit{valid}$.
2. Compute $d_1 = H_z(t)$, set $D = (d_1, d_2)$, and verify that $E \neq D^f$.

5.4 Security Analysis

In this section, we present a number of theorems corresponding to each of the security properties of LASER defined in Section 5.2. In the existing literature, there are mainly two security models for the DAA schemes:(1) simulation based model [15, 5], and (2) game based model [48, 49, 45, 46]. In the simulation based security model, real-ideal paradigm (i.e. the ideal DAA functionality and the real DAA protocol) is considered. Informally speaking, if an attack on the real protocol can be carried out in the ideal model, then the real protocol is implemented securely. This is because the ideal model is secure. The security of the first DAA scheme was proved in the simulation based model [15]. Recently, this model has been refined to universally composable [54].

While the game-based model, or the feature based model, addresses (typically, one-by-one) the expected security properties. The game corresponding to a security property defines what an adversary is supposed to know, observe and compromise. Also, the game defines what is regarded as a successful breach of the specific security property. It is notable that not all the existing DAA schemes using the game based model preserve security. Bernhard et. al. [46] discuss that the revocation needs to be considered in the security model. Additionally, the

tracing algorithm for the DAA signatures need to be formally described. Further, Camenisch et. al. [5] identify specific flaws in the proofs of security of a number of the existing DAA schemes.

Hence, our security model addresses the issues raised in [46, 5] in the following manner.

- In the definition of the correctness, the revocation is considered since the Verify algorithm includes the revocation check procedure.
- The tracing algorithm Identify is defined and is utilized in the security proofs.

5.4.1 Correctness

Theorem 5.1. *LASER satisfies the correctness property.*

Proof. To show that an honest signature can be verified as valid, it is sufficient to show that $\widetilde{R}_{1s}$ and $\widetilde{R}_{2s}$ retrieved in the Verify algorithm are the same as R_{1s} and R_{2s} computed in the Sign algorithm, respectively. Hence, we present the following.

$$
\begin{aligned}
\widetilde{R}_{1s} &= D^{s_{fs}} \cdot E^{-c_{ts}} = D^{r_{fs}} \cdot D^{f \cdot c_{ts}} \cdot E^{-c_{ts}} = R_{1s}, \\
\widetilde{R}_{2s} &= e(h_1, g_2)^{s_{fs}} \cdot e(h_2, g_2)^{s_y} \cdot e(h_3, \omega \cdot g_2^{x_{jk}})^{s_\delta} \cdot e(T, \omega \cdot g_2^{x_{jk}})^{-c_{ts}} \cdot e(g_1, g_2)^{c_{ts}} \\
&= e(h_1, g_2)^{r_{fs}} \cdot e(h_2, g_2)^{r_y} \cdot e(h_3, \omega \cdot g_2^{x_{jk}})^{r_\delta} \cdot \\
&\quad e(h_1, g_2)^{c_{ts} \cdot f} \cdot e(h_2, g_2)^{c_{ts} \cdot y} \cdot e(h_3, \omega \cdot g_2^{x_{jk}})^{c_{ts} \cdot \delta} \cdot e(T, \omega \cdot g_2^{x_{jk}})^{-c_{ts}} \cdot e(g_1, g_2)^{c_{ts}} \\
&= R_{2s} \cdot \Big(e(h_1^f, g_2) \cdot e(h_2^y, g_2) \cdot e(g_1, g_2) \Big)^{c_{ts}} \cdot \Big(e(T, g_2^{\gamma + x_{jk}}) \cdot e(h_3^\delta, g_2^{\gamma + x_{jk}})^{-1} \Big)^{-c_{ts}} \\
&= R_{2s} \cdot e(A_{jk}^{\gamma + x_{jk}}, g_2)^{c_{ts}} \cdot e(T \cdot h_3^{-\delta}, g_2^{\gamma + x_{jk}})^{-c_{ts}} \\
&= R_{2s} \cdot e(A_{jk}, g_2^{\gamma + x_{jk}})^{c_{ts}} \cdot e(A_{jk}, g_2^{\gamma + x_{jk}})^{-c_{ts}} \\
&= R_{2s}.
\end{aligned}
$$

Note that the equality holds since $A_{jk}^{\gamma+x_{jk}} = g_1 \cdot h_1^f \cdot h_2^y$, for any $j \in [1, m_s]$ and $k \in [1, m_a]$, for an honest platform. In addition, the signature can be identified correctly using the Identify algorithm since $E = D^f$. □

5.4.2 User-Controlled Anonymity

Theorem 5.2. *In the random oracle model, LASER is user-controlled anonymous under the $\mathbb{G}_1$-DDH assumption, i.e., if there is an adversary $\mathcal{A}$ that succeeds with a non-negligible probability to break user-controlled anonymity of LASER, then there exists a PPT simulator $\mathcal{B}$ that breaks the $\mathbb{G}_1$-DDH assumption with a non-negligible probability.*

Proof. To show that a signature generated in LASER is anonymous, a simulator $\mathcal{B}$ is constructed to reduce the problem of winning the anonymity game to breaking a $\mathbb{G}_1$-DDH problem. To achieve this, the simulator $\mathcal{B}$ crafts the membership and signing credentials for a specific platform ID^* without knowing its actual `tsk`. In the user-controlled anonymity game, the simulator $\mathcal{B}$ leverages the random oracle to respond to the *Sign* queries. In the *Challenge* phase, the simulator $\mathcal{B}$ crafts the signing credentials such that the simulator $\mathcal{B}$ perfectly simulates the game, and breaks the $\mathbb{G}_1$-DDH assumption. In the following discussion, we provide the details of the proof.

Let $(P, P^a, P^b, P^c) \in \mathbb{G}_1^4$ be the instance of the $\mathbb{G}_1$-DDH problem, where the simulator $\mathcal{B}$ wishes to answer whether $c = a \cdot b$ or $c \xleftarrow{R} \mathbb{Z}_p^*$. The simulator $\mathcal{B}$ sets up a simulated user-controlled anonymity game, and solves the $\mathbb{G}_1$-DDH problem with the help of an adversary $\mathcal{A}$ who wins the user-controlled anonymity game with non-negligible probability.

1. *Setup*: $\mathcal{B}$ runs the Setup algorithm, sets $h_1 = P$, and sends `isk` and `gpk` to the adversary $\mathcal{A}$. $\mathcal{B}$ selects a specific TPM identity ID^*. Afterwards, for all TPMs except the TPM with identity ID^*, $\mathcal{B}$ outputs a TPM's secret key `tsk`, runs GetMemCre, and runs GetSignCre for m_s number of times. For TPM with identity ID^*, $\mathcal{B}$ runs the following two algorithms.

(a) *GetMemCre-ID**: $\mathcal{B}$ sets $I = P^b = h_1^b$. Then, $\mathcal{B}$ generates the `memCre` using the algorithm GetMemCre assuming that `tsk` $= b$.

(b) *GetSignCre-ID**: $\mathcal{B}$ selects $\theta \xleftarrow{R} \mathbb{Z}_p$, and sets $B_j = P^\theta$ and $K_j = (P^b)^\theta = (P^\theta)^b$. $\mathcal{B}$ selects $a_j \xleftarrow{R} \mathbb{Z}_p$, and set $B_j = (b_{1j}, b_{2j})$. $\mathcal{B}$ runs GetSignCre algorithm that `tsk` $= b$. Afterwards, $\mathcal{B}$ backpatches the hash oracle by setting $H_g(a_j) := B_j$.

2. *Queries*:

(a) *Sign*: There are two cases for replying a *Sign* query.

i. When the adversary $\mathcal{A}$ queries a signature of a TPM with identity $ID \neq ID^*$, $\mathcal{B}$ uses the corresponding TPM's secret key `tsk` and `signCre` in the Sign algorithm to generate a signature, and responds with it.

ii. When the adversary $\mathcal{A}$ queries the signature of the TPM with identity ID^*, $\mathcal{B}$ forges the signature, and backpatches the hash oracle to preserve consistency as follows. $\mathcal{B}$ selects $\zeta \xleftarrow{R} \mathbb{Z}_p$, and sets $D = P^\zeta$ and $E = (P^b)^\zeta = (P^\zeta)^b$. $\mathcal{B}$ sets $s_{fs} \xleftarrow{R} \mathbb{Z}_p$, runs the Sign algorithm, and generates the signature assuming that `tsk` $= b$. Then, $\mathcal{B}$ backpatches the hash oracle by setting $H_z(\texttt{gpk} \parallel D \parallel E \parallel T \parallel \widetilde{R}_{1s} \parallel \widetilde{R}_{2s}) := c_{hs}$. $\mathcal{B}$ also backpatches the hash oracle by setting $H_g(t) := D$.

(b) *FetchMemCre*: $\mathcal{B}$ runs the GetMemCre algorithm, and responds with the `memCre`.

(c) *FetchSignCre*: If the adversary $\mathcal{A}$ queries the signing credential of TPM with identity $ID \neq ID^*$, then $\mathcal{B}$ runs the GetSignCre algorithm and responds with the result; otherwise, $\mathcal{B}$ runs GetSignCre $- ID^*$, and responds with the result.

(d) *FetchReg*: $\mathcal{B}$ responds with the registration list `reg`.

(e) *Corrupt*: If the adversary $\mathcal{A}$ queries the secret key of TPM with identity $ID \neq ID^*$, then $\mathcal{B}$ responds with its `tsk`; otherwise, $\mathcal{B}$ quits and outputs `abortion 1`.

3. *Challenge*: The adversary $\mathcal{A}$ submits a message M, and two identities ID_0 and ID_1. If $ID^* \notin \{ID_0, ID_1\}$, then $\mathcal{B}$ quits and outputs `abortion 2`. Otherwise, $\mathcal{B}$ picks

$\phi \xleftarrow{R} \{0,1\}$ such that $ID^* = ID_\phi$, and generates the signature σ_s^* as follows.

(a) $\mathcal{B}$ selects $v, \mu \xleftarrow{R} \mathbb{Z}_p$, and sets $B_j = (P^a)^\mu$ and $K_j = (P^c)^\mu = (P^\mu)^c$. Let $B_j = (b_{1j}, b_{2j})$. $\mathcal{B}$ runs GetSignCre algorithm assuming that $\texttt{tsk} = c/a$. Afterwards, $\mathcal{B}$ backpatches the hash oracle by setting $H_g(v) := B_j$.

(b) $\mathcal{B}$ selects $\zeta \xleftarrow{R} \mathbb{Z}_p$, and sets $D = P^\zeta$ and $E = (P^b)^\zeta = (P^\zeta)^b$. $\mathcal{B}$ generates the signature assuming that $\texttt{tsk} = b$. $\mathcal{B}$ sets $s_{fs} \xleftarrow{R} \mathbb{Z}_p$.

(c) $\mathcal{B}$ backpatches the hash oracle by setting $H_z(\texttt{gpk} \parallel D \parallel E \parallel T \parallel \widetilde{R}_{1s} \parallel \widetilde{R}_{2s}) := c_{hs}$. $\mathcal{B}$ also backpatches the hash oracle by setting $H_g(t) := D$.

4. *Output*: $\mathcal{A}$ outputs $\phi' \in \{0,1\}$ as the guess for ϕ, or aborts. If $\phi = \phi'$, then $\mathcal{B}$ outputs 1, which means that $c = a \cdot b$. Otherwise $\mathcal{B}$ outputs 0, which means that $c \xleftarrow{R} \mathbb{Z}_p$.

There are three cases where $\mathcal{B}$ may abort the game:

1. Backpatch collision: It happens when the backpatched hash element has already been queried, the probability of which is $O(1/p)$, i.e a negligible number.

2. `abortion 1`: Since $\mathcal{A}$ can not corrupt all the TPMs, the probabilty of $\mathcal{B}$ doesn't abort in this case is at least $1/(m_t + q_m)$, where m_t is the total number of platforms in the setup phase, and q_m is the number of *FetchMemCre* queries.

3. `abortion 2`: $\mathcal{B}$ doesn't abort in this case when ID^* is picked in the challenge phase, the probabilty of $\mathcal{B}$ doesn't abort in this case is at least $1/(m_t + q_m)$.

Therefore, we see $\mathcal{B}$ doesn't abort the game in an non-negligible probability. Further, let ϵ be the probability that $\mathcal{A}$ succeeds in breaking the user-controlled-anonymity game, given that $\mathcal{B}$ does not abort during the above simulation. If $c = ab$, then $\mathcal{B}$ simulates the game perfectly, i.e. $Pr[\phi = \phi'] > \epsilon + \frac{1}{2}$. In the other case where $c \xleftarrow{R} \mathbb{Z}_p$, $\mathcal{B}$ doesn't simulate the game perfectly, since in the challenge phase the new entry appended to `reg` isn't generated

using the same `tsk`. In this case $\mathcal{A}$ would abort the game or lose the game. Therefore, $\mathcal{B}$ has probability of at least $\epsilon/2$ in solving the $\mathbb{G}_1$-DDH problem.

□

5.4.3 Traceability

Lemma 5.3. *Suppose an algorithm $\mathcal{A}$ which is given an instance $(\tilde{g}_1, g_2, g_2^{\gamma}, h_1, h_2)$ and m_c tuples $(f_j, y_j, A_{j1}, x_{j1}, A_{j2}, x_{j2}, \cdots A_{jm_a}, x_{jm_a})$, $\forall j \in [1, m_c]$, where $x_{jk} \in \mathbb{Z}_p^*$ $\forall j \in [1, m_c]$, $\forall k \in [1, m_a]$, $\tilde{g}_1, h_1, h_2 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and $\tilde{A}_{jk} = (\tilde{g}_1 \cdot h_1^{f_j} \cdot h_2^{y_j})^{(1/\gamma + x_{jk})}$, forges a tuple (A_*, x_*, f_*, y_*) for some $A_* \in \mathbb{G}_1$, $x_*, f_*, y_* \in \mathbb{Z}_p$ and $x_* \neq x_{jk}$ $\forall i \in [1, m_c], \forall k \in [1, m_a]$ such that $e(A_*, g_2^{\gamma + x_*}) = e(\tilde{g}_1 h_1^{f_*} h_2^{y_*}, g_2)$, then there exists an algorithm $\mathcal{B}$ which solves the q-SDH problem, where $q \geq m_a m_c$.*

Proof. Algorithm $\mathcal{B}$ is given a q-SDH instance represented by $(g_1, \omega_1, \cdots, \omega_q, g_2, g_2^{\gamma})$, where $\omega_i = g_1^{\gamma^i}$, $\forall i \in [1, q]$. $\mathcal{B}$ sets $q = m_c m_a$. The objective of $\mathcal{B}$ is to produce a SDH pair $(g_1^{1/(\gamma+d)}, d)$ for some $d \in \mathbb{Z}_p^*$. For this, $\mathcal{B}$ creates the following framework to interact with the algorithm $\mathcal{A}$.

1. *Setup*: $\mathcal{B}$ does the following.
   (a) Select $m_a m_c$ parameters, $x_{jk} \xleftarrow{R} \mathbb{Z}_p^*$, $\forall j \in [1, m_c], \forall k \in [1, m_a]$.
   (b) Define $\pi_j = \prod_{k=1}^{m_a} (\gamma + x_{jk})$, and $F(\gamma) = \prod_{j=1}^{m_c} \pi_j = \sum_{i=0}^{m_a m_c} \alpha_i \gamma^i$, where $\alpha_0, \alpha_1, \cdots, \alpha_{m_a m_c} \in \mathbb{Z}_p^*$ are the coefficients of the polynomial F with variable γ.
   (c) Compute $\tilde{g}_1 = g_1^{F(\gamma)} = \prod_{i=0}^{m_a m_c} \omega_i^{\alpha_i}$. Note that we denote $\omega_0 = g_1$.
   (d) Select $r_1, r_2 \xleftarrow{R} \mathbb{Z}_p^*$; and compute $h_1 = \tilde{g}_1^{r_1}$, and $h_2 = \tilde{g}_1^{r_2}$.
   (e) Select $f_j, y_j \xleftarrow{R} \mathbb{Z}_p^*$, $\forall j \in [1, m_c]$.
   (f) Define $F_{jk}(\gamma) = F(\gamma)/(\gamma + x_{jk}) = \prod_{j=1, j \neq i}^{m_c} \pi_i \cdot \prod_{l=1, l \neq k}^{m_a} (\gamma + x_{jl}) = \sum_{i=0}^{m_a m_c - 1} a_i \gamma^i$, where $a_0, a_1, \cdots, a_{m_a m_c - 1} \in \mathbb{Z}_p^*$ are the coefficients of the polynomial F_{jk}.

(g) Compute

$$A_{jk} = (\tilde{g}_1 h_1^{f_j} h_2^{y_j})^{\frac{1}{\gamma + x_{jk}}} = g_1^{\frac{F(\gamma)\cdot(1+r_1\cdot f_j + r_2\cdot y_j)}{\gamma + x_{jk}}} = g_1^{F_{jk}(\gamma)\cdot(1+r_1 f_j + r_2 y_j)} = \prod_{i=0}^{m_a m_c - 1} \omega_i^{a_i\cdot(1+r_1\cdot f_j + r_2\cdot y_j)}.$$

(h) Send $(f_j, y_j, A_{j1}, x_{j1}, A_{j2}, x_{j2}, \cdots, A_{jm}, x_{jm})$, and $(\tilde{g}_1, g_2, g_2^{\gamma}, h_1, h_2)$ to $\mathcal{A}$.

Note that with this information, $\mathcal{A}$ or $\mathcal{B}$ has got $m_a m_c$ tuples $(f_j, y_j, A_{jk}, x_{jk})$ such that $e(A_{jk}, g_2^{\gamma} \cdot g_2^{x_{jk}}) = e(\tilde{g}_1 \cdot h_1^{f_j} \cdot h_2^{y_j}, g_2)$.

2. *Output*: $\mathcal{A}$ outputs a forged tuple (f_*, y_*, A_*, x_*), for some $A_* \in \mathbb{G}_1$ and $x_* \neq x_{jk}$, $\forall j \in [1, m_c], \forall k \in [1, m_a]$, such that $e(A_*, g_2^{\gamma} \cdot g_2^{x_*}) = e(\tilde{g}_1 \cdot h_1^{f_*} \cdot h_2^{y_*}, g_2)$.

Having received the forged tuple from $\mathcal{A}$, $\mathcal{B}$ generates a new SDH pair in the following manner.

1. Note that

$$A_* = (\tilde{g}_1 \cdot h_1^{f_*} \cdot h_2^{y_*})^{\frac{1}{\gamma + x_*}} = g_1^{\frac{F(\gamma)\cdot(1+r_1\cdot f_* + r_2\cdot y_*)}{\gamma + x_*}} \tag{5.1}$$

2. Assume that $F(\gamma)\cdot(1 + r_1\cdot f_* + r_2\cdot y_*) = (\gamma + x_*)\cdot F_d(\gamma) + d_*$ for some polynomial $F_d(\gamma) = \sum_{i=0}^{m_a m_c - 1} d_i \cdot \gamma^i$, and constant $d_* \in \mathbb{Z}_p^*$. This means that

$$A_* = g_1^{F_d(\gamma) + \frac{d_*}{\gamma + x_*}}.$$

3. Compute $g_1^{F_d(\gamma)} = \prod_{i=0}^{m_a m_c - 1} \omega_i^{d_i}$.

4. Obtain

$$g_1^{\frac{1}{\gamma + x_*}} = \left(\frac{A_*}{g_1^{F_d(\gamma)}}\right)^{\frac{1}{d_*}}$$

Hence, $\mathcal{B}$ returns the tuple $(g_1^{\frac{1}{\gamma + x_*}}, x_*)$ as the solution to the submitted instance of the SDH problem. □

Theorem 5.4. *In the random oracle model, LASER is traceable under q-SDH assumption, i.e., if there is an adversary $\mathcal{A}$ that succeeds with a non-negligible probability to break traceability of LASER, then there exists a PPT simulator $\mathcal{B}$ that breaks the q-SDH assumption with a non-negligible probability, where $q = m_a \cdot m_c + m_t$, m_t is the number of TPMs, m_c is the number of signing credentials, and m_a is the number of alias tokens for each signing credential.*

Proof. To show LASER is traceable, the simulator $\mathcal{B}$ is constructed to reduce a successful forgery (i.e., wining traceability game) to breaking the q-SDH assumption. The simulator $\mathcal{B}$ utilizes Lemma 5.3 to create q valid signing credentials $\mathsf{signCre}$ without knowing the issuer's secret key $\mathtt{isk}$. Upon a successful forgery, the simulator $\mathcal{B}$ rewinds the algorithm, and extracts a new valid $\mathsf{signCre}$. Finally, using Lemma 5.3, the simulator $\mathcal{B}$ breaks the q-SDH assumption.

Let $(g_1, \omega_1, \cdots, \omega_q, g_2, g_2^{\gamma})$ be the instance of the q-SDH assumption, where $\omega_i = g_1^{\gamma^i}$, $\forall i \in [1, q]$, and $q = m_a m_c + m_t$. The simulator $\mathcal{B}$ wishes to construct a tuple $(d, g_1^{\frac{1}{\gamma+d}})$. To achieve this, $\mathcal{B}$ sets up a simulated traceability game as follows.

1. *Setup*: $\mathcal{B}$ does the following.
   (a) Select $m_a m_c$ alias tokens, $x_{jk} \xleftarrow{R} \mathbb{Z}_p^*$, $\forall j \in [1, m_c], \forall k \in [1, m_a]$.
   (b) Select m_t parameters, $z_j \xleftarrow{R} \mathbb{Z}_p^*$, $\forall j \in [1, m_t]$.
   (c) Define $\pi_j = \prod_{k=1}^{m_a}(\gamma + x_{jk})$, and $F(\gamma) = \left(\prod_{j=1}^{m_c} \pi_j\right) \cdot \left(\prod_{j=1}^{m_t}(\gamma + z_j)\right) = \sum_{i=0}^{m_a m_c + m_t} \alpha_i \gamma^i$, where $\alpha_0, \alpha_1, \cdots, \alpha_{m_a m_c + m_t} \in \mathbb{Z}_p^*$ are the coefficients of the polynomial F with variable γ.
   (d) Compute $\tilde{g}_1 = g_1^{F(\gamma)} = \prod_{i=0}^{m_a m_c + m_t} \omega_i^{\alpha_i}$. Note that we denote $\omega_0 = g_1$.
   (e) Select $r_1, r_2 \xleftarrow{R} \mathbb{Z}_p^*$; and compute $h_1 = \tilde{g}_1^{r_1}$, and $h_2 = \tilde{g}_1^{r_2}$.

   Afterwards, $\mathcal{B}$ runs the Setup algorithm assuming $\mathtt{isk} = \gamma$, and $\tilde{g}_1$ is the generator of $\mathbb{G}_1$.

2. *Queries*: $\mathcal{A}$ can query $\mathcal{B}$ about the following.

   (a) *Sign*: $\mathcal{B}$ runs Sign algorithm, and responds with the result.

   (b) *FetchMemCre*: For TPM with identity j, $\mathcal{B}$ sets its secret key $\mathsf{tsk} = f$, and constructs its `memCre` as follows.

      i. Select $\rho \xleftarrow{R} \mathbb{Z}_p^*$.

      ii. Define $F_j(\gamma) = F(\gamma)/(\gamma + z_j) = \prod_{i=1}^{m_c} \pi_i \cdot \prod_{l=1, l\neq j}^{m_t} (\gamma + z_l) = \sum_{i=0}^{m_a m_c + m_t - 1} a_i \gamma^i$, where $a_0, a_1, \cdots, a_{m_a m_c + m_t - 1} \in \mathbb{Z}_p^*$ are the coefficients of the polynomial F_j.

      iii. Compute

      $$J = (\tilde{g}_1 h_1^f h_2^\rho)^{\frac{1}{\gamma + z_j}} = g_1^{\frac{F(\gamma)\cdot(1+r_1\cdot f + r_2\cdot\rho)}{\gamma + z_j}} = g_1^{F_j(\gamma)\cdot(1+r_1 f + r_2 y_j)} = \prod_{i=0}^{m_a m_c + m_t - 1} \omega_i^{a_i\cdot(1+r_1\cdot f + r_2\cdot\rho)}.$$

   (c) *FetchSignCre*: $\mathcal{B}$ constructs $\mathtt{signCre_j}$ for a TPM with $\mathtt{tsk} = f$ as follows:

      i. Select $y_j \xleftarrow{R} \mathbb{Z}_p^*$, and m_a unassigned alias tokens, $\{x_{j1}, x_{j2}, \cdots, x_{jm_a}\}$, and set `signCre`.

      ii. Define $F_{jk}(\gamma) = F(\gamma)/(\gamma + x_{jk}) = \prod_{i=1, i\neq j}^{m_c} \pi_i \cdot \prod_{l=1, l\neq k}^{m_a} (\gamma + x_{jl}) \cdot \prod_{i=1}^{m_t} (\gamma + z_i) = \sum_{i=0}^{m_a m_c + m_t - 1} a_i \gamma^i$, where $a_0, a_1, \cdots, a_{m_a m_c + m_t - 1} \in \mathbb{Z}_p^*$ are the coefficients of the polynomial F_{jk}.

      iii. Compute

      $$A_{jk} = (\tilde{g}_1 h_1^f h_2^{y_j})^{\frac{1}{\gamma + x_{jk}}} = g_1^{\frac{F(\gamma)\cdot(1+r_1\cdot f + r_2\cdot y_j)}{\gamma + x_{jk}}} = g_1^{F_{jk}(\gamma)\cdot(1+r_1 f + r_2 y_j)}$$
      $$= \prod_{i=0}^{m_a m_c + m_t - 1} \omega_i^{a_i\cdot(1+r_1\cdot f + r_2\cdot y_j)}.$$

      Further, $\mathcal{B}$ compute a_j, b_{2j}, and K_j, and append $(a_j, b_{2j}, K_j, x_{j1}, \cdots, x_{jm_a})$ to the database registry `reg`.

   (d) *FetchReg*: $\mathcal{B}$ responds with the database registry `reg`.

   (e) *Corrupt*: $\mathcal{B}$ responds with corresponding `tsk`.

3. *Output*: Finally, if $\mathcal{A}$ is successful, it outputs a forged signature σ_s on a message M using am alias token x_* such that it is not identified as any TPM.

The framework succeeds whenever $\mathcal{A}$ succeeds. Hence, $\mathcal{B}$ obtains a successful forgery with non-negligible probability. Then, $\mathcal{B}$ rewinds the framework to obtain two forged signatures on the same message, where the commitments are the same, but the challenges and responses are different. By the forking lemma [4, 91], the probability of successfully achieving this is non-negligible, and $\mathcal{B}$ extracts (A_*, f_*, y_*) encoded in the forged signatures. Further, $\mathcal{B}$ obtains an SDH pair from (A_*, x_*) using the technique discussed in Lemma 5.3. Therefore, $\mathcal{B}$ breaks q-SDH assumption with non-negligible probability.

□

5.4.4 Non-frameability

Theorem 5.5. *In the random oracle model, LASER is non-frameable under the $\mathbb{G}_1$-DL assumption, i.e., if there is an adversary $\mathcal{A}$ that succeeds with a non-negligible probability to break non-frameability of LASER, then there exists a PPT simulator $\mathcal{B}$ that breaks the $\mathbb{G}_1$-DL assumption with a non-negligible probability.*

Proof. Let (P, P^μ) be a $\mathbb{G}_1$-DL instance for some $\mu \in \mathbb{Z}_p$. The objective of $\mathcal{B}$ is to compute μ. To achieve this, $\mathcal{B}$ sets up a simulated non-frameability game as follows.

1. *Setup*: $\mathcal{B}$ runs the Setup algorithm, sets $h_1 = P$, and sends `isk` and `gpk` to the adversary $\mathcal{A}$. $\mathcal{B}$ selects a specific TPM identity ID^*. Afterwards, for each TPM except the TPM with identity ID^*, $\mathcal{B}$ outputs a TPM's secret key `tsk`, runs GetMemCre, and runs GetSignCre for m_s number of times. For TPM with identity ID^*, $\mathcal{B}$ runs the following two algorithms.

   (a) *GetMemCre-ID^**: $\mathcal{B}$ sets $I = P^\mu = h_1^\mu$. Then, $\mathcal{B}$ generates the `memCre` using the algorithm GetMemCre assuming that $\texttt{tsk} = \mu$.

   (b) *GetSignCre-ID^**: $\mathcal{B}$ selects $\theta \xleftarrow{R} \mathbb{Z}_p$, and sets $B_j = P^\theta$ and $K_j = (P^\mu)^\theta = (P^\theta)^\mu$. $\mathcal{B}$ selects $a_j \xleftarrow{R} \mathbb{Z}_p$, and sets $B_j = (b_{1j}, b_{2j})$. $\mathcal{B}$ generates the $\texttt{signCre}_\texttt{j}$ using

GetSignCre assuming that $\texttt{tsk} = \mu$. Afterwards, $\mathcal{B}$ backpatches the hash oracle by setting $H_g(a_j) := B_j$.

2. *Queries*:

   (a) *Sign*: There are two cases for replying a *Sign* query.

      i. When the adversary $\mathcal{A}$ queries a signature of a TPM with identity $ID \neq ID^*$, $\mathcal{B}$ uses the corresponding TPM's secret key $\texttt{tsk}$ and $\texttt{signCre}$ in the Sign algorithm to generate a signature, and responds with it.

      ii. When the adversary $\mathcal{A}$ queries the signature of the TPM with identity ID^*, $\mathcal{B}$ forges the signature, and backpatches the hash oracle to preserve consistency as follows. $\mathcal{B}$ selects $\zeta \xleftarrow{R} \mathbb{Z}_p$, and sets $D = P^{\zeta}$ and $E = (P^{\mu})^{\zeta} = (P^{\zeta})^{\mu}$. $\mathcal{B}$ sets $s_{fs} \xleftarrow{R} \mathbb{Z}_p$, runs the Sign algorithm, and generates the signature assuming that $\texttt{tsk} = \mu$. Then, $\mathcal{B}$ backpatches the hash oracle by setting $H_z(\texttt{gpk} \parallel D \parallel E \parallel T \parallel \widetilde{R}_{1s} \parallel \widetilde{R}_{2s}) := c_{hs}$. $\mathcal{B}$ also backpatches the hash oracle by setting $H_g(t) := D$.

   (b) *FetchMemCre*: $\mathcal{B}$ runs GetMemCre, and responds with the result.

   (c) *FetchSignCre*: If the adversary $\mathcal{A}$ queries the signing credential of TPM with identity $ID \neq ID^*$, then $\mathcal{B}$ runs GetSignCre and responds with the result. Otherwise, $\mathcal{B}$ runs GetSignCre-ID^* and responds with the result.

   (d) *FetchReg*: $\mathcal{B}$ responds with the database registry $\texttt{reg}$.

   (e) *Corrupt*: If the adversary $\mathcal{A}$ queries the secret key of TPM with identity $ID \neq ID^*$, then $\mathcal{B}$ responds with its $\texttt{tsk}$; otherwise, $\mathcal{B}$ aborts and quits the game.

3. *Output*: If $\mathcal{A}$ is successful, it outputs a forged signature σ_s on a message M, and the TPM identity $\widetilde{ID}$ where σ_s is identified as $\widetilde{ID}$.

If $\widetilde{ID} \neq ID^*$, $\mathcal{B}$ aborts the game. There are three cases where $\mathcal{B}$ aborts the game.

1. Backpatch collision: It happens when the backpatched hash element has already been queried, the probability of which is $O(1/p)$.

2. Corruption of ID^*: Since $\mathcal{A}$ can not corrupt all the TPMs, the probabilty that $\mathcal{B}$ does not abort in this case is at least $1/(m_t + q_m)$, where m_t is the total number of platforms in the setup phase, and q_m is the number of *FetchMemCre* queries.

3. Not forging ID^*: The probabilty that $\mathcal{B}$ doesn't abort in this case is at least $1/(m_t+q_m)$.

In either case, the probability that $\mathcal{B}$ does not abort the game is non-negligible. Hence, $\mathcal{B}$ does not abort the game with non-negligible probability.

If $\mathcal{B}$ does not abort the game, it rewinds the framework to obtain two forged signatures on the same message, where the commitments are the same, but the challenges and responses are different. By forking lemma[4, 91], the probability of successfully achieving this is non-negligible, and $\mathcal{B}$ extracts (A_*, f_*, y_*) encoded in the forged signatures. Then, f_* is equal to μ since the forged signature can be identified as TPM ID^*. Hence. $\mathcal{B}$ breaks $\mathbb{G}_1$-DL assumption with non-negligible probability.

□

5.5 Performance Evaluation

In this section, we evaluate the computational and communication overheads of LASER, and compare LASER's performance with two schemes in the prior art: (1) enhanced privacy ID from bilinear pairing (BL-EPID) [3]; and (2) the scheme proposed by Camenisch, Drijvers and Lehmann (CDL-EPID) [5]. We select these two schemes since they employ the notion of revoking a platform based on its malicious signature (i.e., signature based revocation). Note that, in BL-EPID, the operations performed at the platform are not divided between the TPM and the host. Hence, in BL-EPID, we consider that all the operations at the platform are performed at the host.

Table 5.1: Number of the off-line computational operations in the existing DAA schemes.

	BL-EPID	**CDL-EPID**
TPM	$3E_{G_1}^2, 1E_{G_2}\ 2B_M$	$2E_{G_1}$
Host		$1E_{G_1},\ 1E_{G_2},\ 2B_M$

Table 5.2: Number of the off-line computational operations in LASER.

	LASER	
	GetMemCre	GetSignCre
TPM	$2E_{G_1}$	$(3+3m_r)E_{G_1}$
Host	$1E_{G_1},\ 1E_{G_2},\ 2B_M$	$(4+m_r)E_{G_1},\ 2m_rE_{G_1}^2,\ 1E_{G_1}^3,\ m_aE_{G_2},\ 1E_{G_T},\ (2+m_a)B_M$

Further, we assume symmetric 128-bit security level, which provides approximately the same level of security as an RSA signature with a modulus size of 3072 bits. To achieve the same security strength in the elliptic curve cryptosystem, we utilize Barreto-Naehrig (BN) curves with embedding degree 12 where the lengths of an element in $\mathbb{Z}_p^*$, $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are 256 bits, 512 bits 1024 bits, and 3072 bits [96]. The internal is constructed on the curve of the form $y^2 = x^3 + 3$. Specifically, we utilize the "Type F" internal described in pairing-based cryptography (PBC) library available at [95].

5.5.1 Computational Overhead

Analytical Results

We compare the computational cost in LASER with the benchmarks—viz., BL-EPID and CDL-EPID. We divide the operations at the TPM, the host, and the verifier into two classes—(1) off-line, and (2) on-line. The operations which need to be performed in real time while generating and verifying a signature are classified as on-line operations. On the other hand, all the operations which can be pre-computed or stored, and do not need to be

Table 5.3: Number of the on-line computational operations in the existing DAA schemes.

	BL-EPID	**CDL-EPID**
TPM	$4E_{G_1},\ 2m_rE^2_{G_1},\ 1E^3_{G_T},\ 1B_M$	$(3+3m_r)E_{G_1}$
Host		$(1+m_r)E_{G_1},\ (1+2m_r)E^2_{G_1},\ 1E_{G_T},\ 1B_M$
Verifier	$(1+m_r)E^2_{G_1},\ m_rE^3_{G_1},\ 1E^2_{G_2},\ 1E^4_{G_T},\ 1B_M$	$(1+m_r)E^2_{G_1},\ m_rE^3_{G_1},\ 1E^2_{G_2},\ 1E^4_{G_T},\ 1B_M$

Table 5.4: Number of the on-line computational operations in LASER.

	LASER
TPM	$3E_{G_1}$
Host	$1E_{G_1},\ 1E^2_{G_1},\ 1E_{G_T},\ 1B_M$
Verifier	$2E^2_{G_1},\ 1E_{G_2},\ 1E^3_{G_T},\ 1B_M$

generated in real time are classified as off-line operations. Note that the computational overhead at the platform is computed by the sum of the computational overheads at the TPM and the host. Also, in LASER, the total computational overhead at the platform is computed by summing the computational overheads in GetMemCre and GetSignCre at the TPM and the host. Here, we consider only the most computationally expensive operations—i.e., exponentiation in $\mathbb{G}_1$, exponentiation in $\mathbb{G}_2$, exponentiation in $\mathbb{G}_T$, and bilinear mapping. In the following discussion, k number of j-multi-exponentiations in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are represented by $kE^j_{G_1}$, $kE^j_{G_2}$ and $kE^j_{G_T}$, respectively. The notation kB_M represents k bilinear mappings. The time taken to perform all other operations, e.g., multiplication, addition, inverse, binary search, etc., is significantly smaller when compared to the time taken to compute an exponentiation in $\mathbb{G}_1$, and hence are ignored in the following analysis. Also, we do not consider the computational overhead at the issuer since the issuer is assumed to be significantly more powerful than the platform and the verifier in terms of the computation power and memory.

Off-line The off-line operations include the computations at the TPM and the host for requesting and obtaining the platform's membership and/or signing credentials. Tables 5.1 and 5.2 present the number of computationally expensive off-line operations performed by each entity in the three DAA schemes. In Table 5.1, we observe that the computational complexity of the off-line operations at the platform in BL-EPID and CDL-EPID is O(1) with respect to the number of revoked signing credentials m_r. However, in Table 5.2, we observe that the total computational complexity of the off-line operations at the platform in LASER is O(m_r). Also, we note that the total computational complexity of the platform in LASER is O(m_a) with respect to the number of alias tokens per signing credential m_a. Further, the off-line computational complexity at the platform increases by O(m_s) in LASER because GetSignCre (i.e., the protocol to obtain the signing credential) is performed m_s number of times to obtain m_s signing credentials.

On-line The on-line operations include the computations at the platform for generating the signature, and the computations at the verifier for verifying the signature. Tables 5.3 and 5.4 present the number of computationally expensive on-line operations performed by the TPM, the host and the verifier in the three DAA schemes. We observe that the computational complexity of the on-line operations at the platform is O(m_r) in BL-EPID and CDL-EPID as compared to O(1) in LASER with respect to the number of revoked signing credentials m_r. Also, the computational complexity of the on-line operations at the verifier is O(m_r) in BL-EPID and CDL-EPID.

Simulation Results

Here, we present the computational overheads in LASER in terms of running time. We also evaluate the performance of LASER relative to that of CDL-EPID. We utilize a PC system which is a Macbook Pro 11,1 with 3.0 GHz Intel Duo Core i7 CPU, to measure the running time for the most computationally expensive operations—i.e., E_{G_1}, E_{G_2}, E_{G_T}, and B_M. By

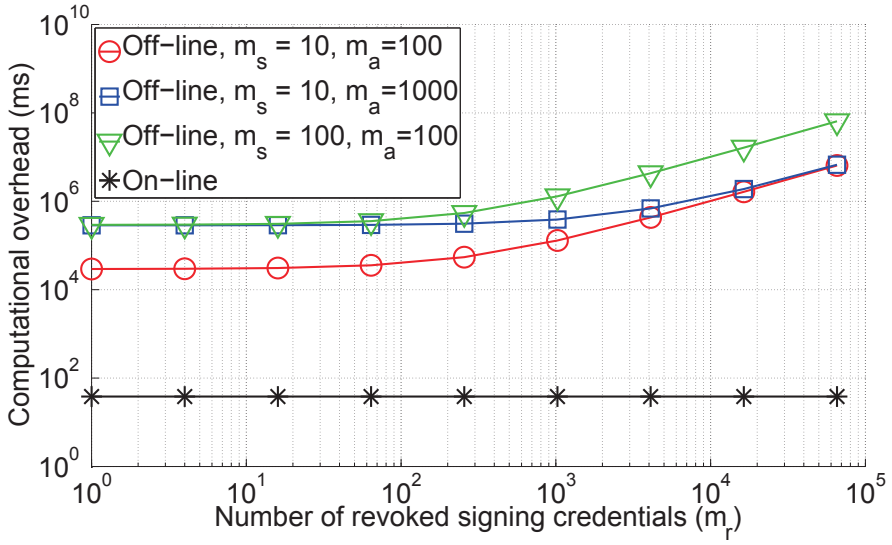


Figure 5.4: Computational overhead at the platform in LASER with different m_s and m_a vs. the number of revoked signing credentials.

Table 5.5: Comparison of the running time (in ms) of different operations in DAA schemes.

Operation	E_{G_1}	$E^2_{G_1}$	$E^3_{G_1}$	E_{G_2}	$E^2_{G_2}$	$E^3_{G_2}$	E_{G_T}	$E^3_{G_T}$	B_M
Time	1.213	1.671	1.816	2.126	2.881	3.204	4.699	7.813	25.828

averaging over 1000 iterations of each of the operations, we obtain the running time of the operations as shown in Table 5.5. From Tables 5.4, we note that the time taken for the on-line signature generation at the platform in LASER is around 37 ms, and the time taken for the on-line signature verification at the verifier in LASER is around 39 ms. Further, using the Tables 5.1, 5.2, 5.3 and 5.4, we generate the plots presented in Figures 5.4 and 5.5.

In Figure 5.4, we present the off-line and on-line computational overheads at the platform in LASER with different values of m_s and m_a vs. the number of revoked signing credentials m_r. In the figure, we observe that as m_s, m_a or m_r increases, the off-line computational overhead increases. Note that for lower values of m_r, the computational overhead is mainly dominated by the values of m_s and m_a. Further, there is no effect of m_r, m_s or m_a on the on-line computational overhead at the platform in LASER.

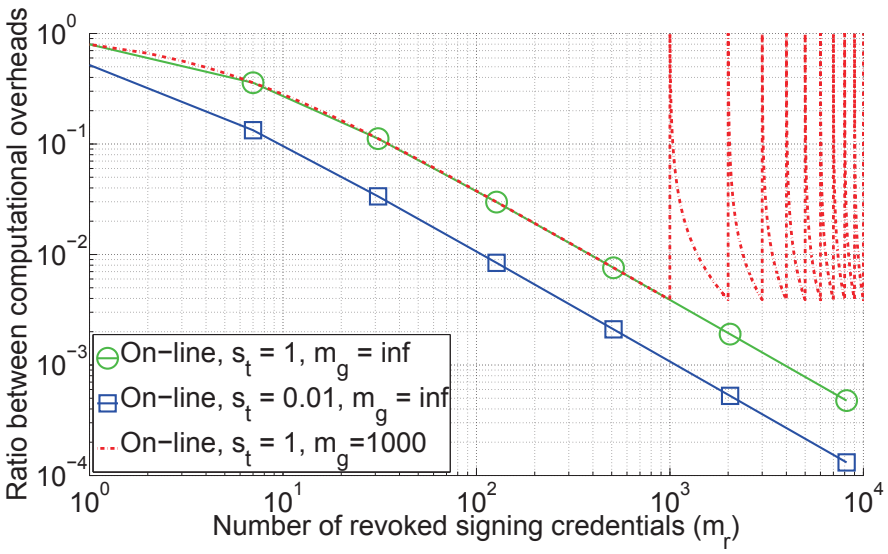


Figure 5.5: Ratio between the computational overhead in LASER (with $m_s = 10$ and $m_a = 100$) and the computational overhead in CDL-EPID vs. the number of revoked signing credentials.

We note that in practice the computational capability of a typical TPM is significantly lesser than that of a typical host. We assume that the ratio between the computational capability of the TPM and that of the host is represented by s_t. To mitigate the high computational overhead in EPID, Brickell and Li in [16], and Chen and Li in [28] discuss a rekeying or a group resetting scheme. In the group resetting scheme, whenever the number of revoked signing credentials increases above a threshold value, represented by m_g, the group can be reset, and new credentials are obtained by each of the platforms in the network. Note that for resetting a group, each platform needs to re-establish a communication link with the issuer, obtain the credentials, and make corresponding computations. Hence, this may not be a feasible solution for a large network with a large number of platforms in the group placed in remote locations. Nevertheless, in this paper, we illustrate the advantage of LASER over the CDL-EPID with and without group resetting scheme.

Figure 5.5 presents the ratio between the on-line computational overhead at the platform in LASER (with $m_s = 10$ and $m_a = 100$) and that in CDL-EPID for different values

of s_t and m_g. We observe that when $m_g = \infty$ (i.e., CDL-EPID without group resetting scheme), the ratio between the on-line computational overhead at the platform in LASER and that in CDL-EPID decreases significantly when m_r increases. For example, with 1000 revoked signing credentials, the ratio is around 0.004. This means that LASER is 250 times more efficient than CDL-EPID in terms of the on-line computational overhead. Further, as the ratio of the computational ability between the TPM and that in host, s_t, decreases, the ratio decrease, and the efficiency of LASER relative to CDL-EPID increases. Further, in this figure, when $m_g = 1000$, we observe that the ratio corresponding to the on-line computational overhead at the platform decreases upto 0.004. This means that when the group is reset in CDL-EPID, the average ratio between the on-line computational overhead at the platform in LASER and that in CDL-EPID is 0.008. Hence, LASER is 125 times more efficient than CDL-EPID with the group resetting scheme in terms of the on-line computational overhead. Note that this significant efficiency is achieved at the cost of higher off-line computational overhead in LASER than that in CDL-EPID. In any case, this trade-off is very advantageous because the on-line operations are computed significantly more often than the off-line operations.

5.5.2 Communication Overhead

Analytical Results

Here, we compare the communication overheads in LASER with those in BL-EPID and CDL-EPID. We divide the communications at the platform and the verifier into two classes—(1) off-line, and (2) on-line. The communications which need to be performed in real time for sending and receiving a signature are classified as on-line communications. On the other hand, all the communications which can be pre-shared and stored, and do not need to be performed in real time are classified as off-line communications. In the following discussion, the lengths of an element in $\mathbb{G}_1$, and $\mathbb{Z}_p^*$ are represented by L_{G_1}, and L_{Zp}, respectively.

Table 5.6: Comparison of the number of parameters in the off-line communication in the DAA schemes.

	BL-EPID	**CDL-EPID**	**LASER**	
			GetMemCre	GetSignCre
platform-issuer-sig	$1L_{G_1},\ 3L_{Zp}$	$1L_{G_1},\ 3L_{Zp}$	$1L_{G_1}, 3L_{Zp}$	$(3+m_r)L_{G_1}, (10+4m_r)L_{Zp}$
issuer-platform-cre	$1L_{G_1},\ 2L_{Zp}$	$1L_{G_1},\ 2L_{Zp}$	$1L_{G_1},\ 2L_{Zp}$	$m_aL_{G_1},\ (1+m_a)L_{Zp}$
issuer-platform-rev	$2m_rL_{G_1}$	$2m_rL_{G_1}$	$m_rL_{G_1},\ 2m_rL_{Zp}$	
issuer-verifier-rev	$2m_rL_{G_1}$	$2m_rL_{G_1}$	$m_rm_aL_{Zp}$	

Table 5.7: Comparison of the number of parameters in the on-line communication in the DAA schemes.

	BL-EPID	**CDL-EPID**	**LASER**
platform-verifier-sig	$(3+m_r)L_{G_1},\ (5+3m_r)L_{Zp}$	$(2+m_r)L_{G_1},\ (6+4m_r)L_{Zp}$	$2L_{G_1},\ 8L_{Zp}$

Off-line The off-line communication overhead includes the communication from the platform to the issuer for sending the signatures with requests for the membership and/or signing credentials (represented by *platform-issuer-sig*). It also includes the communication from the issuer to the platform for sending the membership and/or signing credentials (represented by *issuer-platform-cre*), and the revocation list (represented by *issuer-platform-rev*). Further, it includes the communication from the issuer to the verifier for sending the revocation list (represented by *issuer-verifier-rev*). Table 5.6 presents the number of off-line elements communicated between the entities in the three DAA schemes. Here, the communication overhead at the platform in LASER is computed by summing the communication overheads in GetMemCre and GetSignCre. In Table 5.6, we observe that the complexity of the total communication overhead at the platform and the verifier are O(m_r) in BL-EPID, CDL-EPID, and LASER.

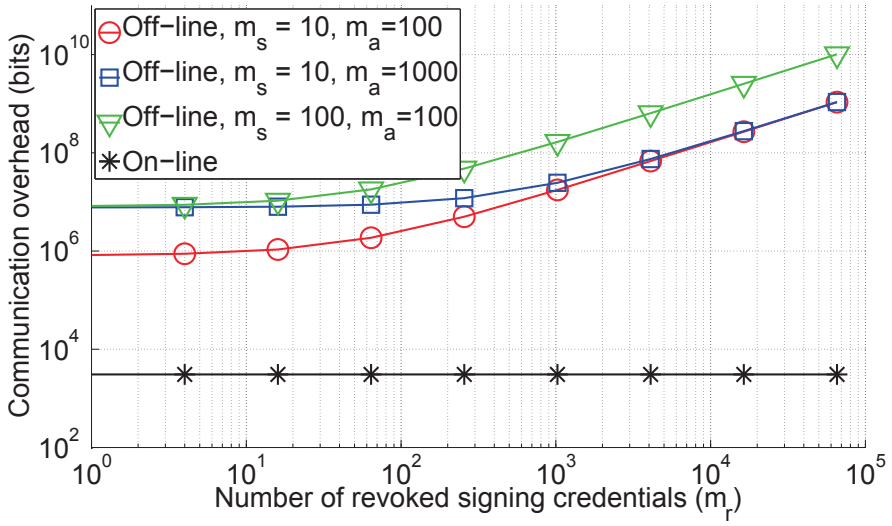


Figure 5.6: Communication overhead at the platform in LASER (with $m_a = 100$ and different m_s) vs. the number of revoked signing credentials.

On-line The on-line communication overhead includes the communication between the platform and the verifier for communicating the signature (represented by *platform-verifier-sig*). Table 5.7 presents the number of on-line elements communicated by each entity in the three DAA schemes. We observe that the communication overhead of the platform and the verifier is O(m_r) in BL-EPID and CDL-EPID as compared to O(1) in LASER.

Simulation Results

Here, we present the communication overheads in LASER in terms of bits. We also evaluate the performance of LASER with respect to CDL-EPID. Recall that for the selected BN curve, $L_{G_1} = 512$ bits, and $L_{Zp} = 256$ bits. Using these values in Tables 5.6 and 5.7, we generate the plots presented in Figures 5.6 and 5.7. Here, we do not consider the communication overhead at the issuer since the issuer is assumed to be significantly more powerful than the platform and the verifier in terms of the communication resources.

We present the off-line and on-line computational overheads at the platform in LASER with

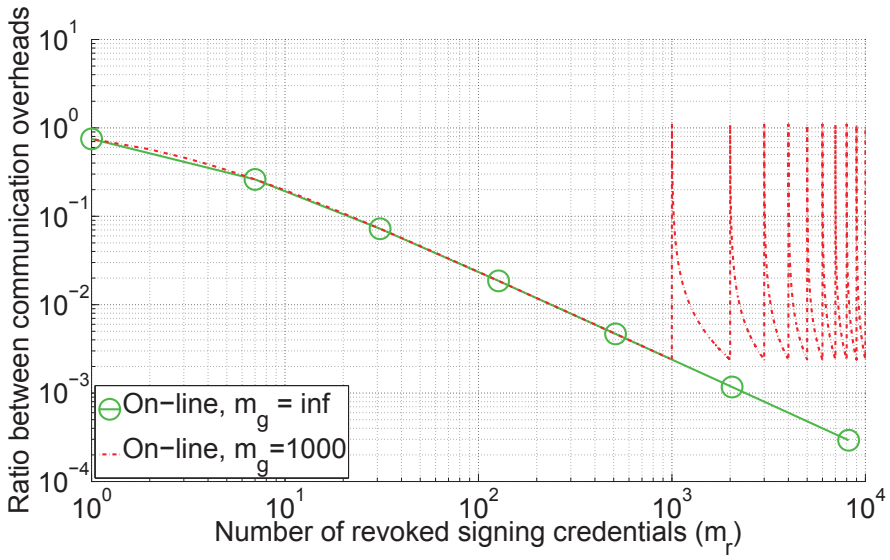


Figure 5.7: Ratio between the communication overhead in LASER (with $m_s = 10$ and $m_a = 100$) and the communication overhead in CDL-EPID vs. the number of revoked signing credentials.

different value of m_s and m_a in Figure 5.6. In this figure, we observe that as m_s, m_a or m_r increases, the off-line communication overhead at the platform increases. However, there is no effect on the on-line communication overhead. Specifically, the length of a signature in LASER is 3072 bits.

Figure 5.7 illustrates the performance of LASER (with $m_s = 10$ and $m_a = 100$) relative to CDL-EPID in terms of the communication overhead. In this figure, we observe that the ratio between the on-line communication overhead at the platform in LASER and that in CDL-EPID, when $m_g = \infty$, decreases significantly when m_r increases. For example, with 1000 revoked signing credentials, the ratio between the on-line communication overhead at the platform in LASER and that in CDL-EPID is around 0.0025. This means that LASER is 400 times more efficient than CDL-EPID in terms of the on-line communication overhead. Also, in this figure, when $m_g = 1000$, we observe that the ratio corresponding to the on-line communication overhead at the platform decreases upto 0.0025. This means that if the group in CDL-EPID is reset after $m_g = 1000$ revoked credentials, the average ratio between

the on-line communication overhead at the platform in LASER and that in CDL-EPID is around 0.005. Hence, LASER is 200 times more efficient than CDL-EPID with the group resetting scheme in terms of the on-line communication overhead. Again, note that this significant efficiency is achieved at the cost of higher off-line communication overhead in LASER than that in CDL-DAA. In any case, this trade-off is very advantageous because the on-line communication occurs significantly more often than the off-line communication.

5.6 Implementation

In this section, we present the implementation results generated using a Macbook Pro 3.0 GHz Intel Duo Core i7 CPU, and a Raspberry Pi 3 board with 1.2 GHz 64-bit quad-core ARMv8 CPU. The OpenSSL and the PBC libraries are leveraged to prototype LASER and CDL-EPID in C programming language [95]. The BN-256 curve which is standardized for DAA by the TCG is utilized [97]. For the implementation, the IBM Trusted Software Stack (TSS) for TPM 2.0 [98], and the software TPM emulator prototyped by IBM [99] are utilized. Note that the emulator has the same programming interface as a physical TPM. Hence, ideally, a software code written for the TPM emulator can be ported to be used with a physical TPM.

A detailed discussion of the TPM 2.0 commands needed to implement DAA applications can be found in [58]. The commands needed for the computations performed in the implementation of LASER are `CreatePrimary`, `Commit`, and `Sign`. The GetMemCre algorithm requires the TPM's secret key to be generated from the `DAAseed` on the TPM using `CreatePrimary`. In the emulator, a signature is generated in two parts using the `Commit` and `Sign` commands. In the GetSignKey operation, the signatures $(\sigma_0, \sigma_1, \cdots \sigma_{m_r})$ are generated using the `Commit` commands followed by the `Sign` commands. In LASER, the Sign algorithm utilizes a single call to the `Commit` and `Sign` commands to generate the signature σ_s on the message M. The individual commands can be run using `TSS_Execute` from the TSS after setting the

Table 5.8: Comparison of the running time (in ms) of the signature generation in the DAA schemes.

			CDL-EPID	**LASER**
Experiment-1	Laptop	TPM	166,956.54	166.43
		Host	30,224.81	74.35
Experiment-2	Pi board	TPM	193,415.10	186.33
	Laptop	Host	29,244.24	65.43

appropriate parameters.

With the software code for the host and the TPM emulator, two experiments are conducted assuming that the number of revoked signing credentials, $m_r = 1000$. In the first experiment, the host as well as the TPM are implemented on the Macbook Pro laptop. In the second experiment, the laptop is utilized to run the host, and the Pi board is utilized to run the TPM. In both the experiments, the host communicates with the emulator over a local network. The results obtained in these experiments are presented in Table 5.8. From this table, it can be inferred that the signature generation in LASER is around 798 times and 865 times more efficient than that in CDL-EPID in the first experiment and the second experiment, respectively.

A typical physical TPM (which is available in the market today) has a processing speed of approximately 33 MHz [100]. Hence, it can be estimated that the computational capability of the laptop used in the first experiment is about 93 times more than that of a physical TPM, and the computational capability of the Pi board in the second experiment is about 37 times more than that of a physical TPM. It can be claimed that an implementation with a physical TPM will also demonstrate that LASER has a significant advantage over CDL-EPID in terms of the on-line computational overhead.

5.7 Summary

In this chapter, a novel DAA scheme called Lightweight Anonymous attestation Scheme with Efficient Revocation (LASER) is proposed. It is shown that the revocation is the primary performance bottleneck of modern DAA schemes and that the existing schemes do not scale well to large networks because of the high computational and communication costs corresponding to their revocation check procedures. By using the novel concept of user-controlled unlinkability, LASER manages to significantly reduce the computational and communication burden of the on-line protocol (i.e., signature generation and verification) at the cost of increased computational and communication overheads of the off-line protocol (i.e., obtaining the membership and signing keys/credentials).

Chapter 6

P-DSA: Precoded Duobinary Signaling for Authentication

In this chapter, we discuss the pulse shaping scheme, called *Precoded Duobinary Signaling* (P-DS). The core idea of P-DS is to introduce a controlled amount of ISI in the transmitted pulses, and change the detection procedure at the receiver to cancel out the ISI [101]. Further, we identify the redundancy inserted into the message signal due to P-DS, and utilize the inherent redundancy as the underlying mechanism to propose a novel intended receiver based authentication (IRA) scheme, called *Precoded Duobinary Signaling for Authentication* (P-DSA) to embed the authentication signal into the message signal. We show that P-DSA do not suffer from the drawbacks of the blind signal superposition approach. Our results indicate that P-DSA outperforms the prior art in terms of the considered performance criteria. We implement P-DSA on Universal Software Radio Peripheral (USRP) radio boards, and verify the validity of the simulation results through indoor experiments.

The rest of the chapter is organized as follows. We present the model used for P-DSA in Section 6.1, and describe the technical background in Section 6.2. We discuss the proposed scheme in Section 6.3, and analyze its error performance in Section 6.4. We evaluate the proposed scheme by comparing with the prior art in Section 6.5. We discuss a prototype

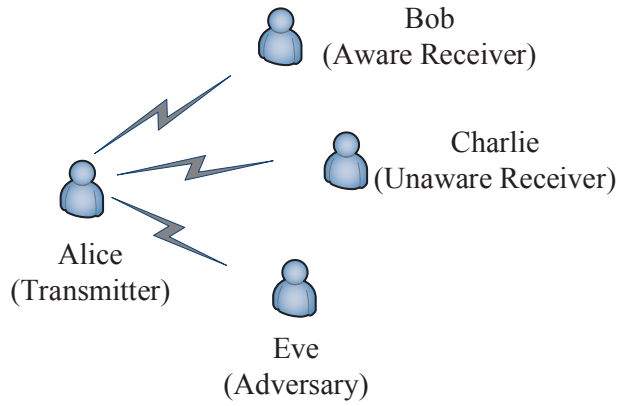


Figure 6.1: Authentication scenario for P-DSA.

implementation of the proposed scheme in Section 6.6. Section 6.7 concludes the chapter by highlighting the main contributions.

6.1 Model and Assumptions

We assume an authentication scenario illustrated in Figure 6.1. In this scenario, Alice, Bob, Charlie and Eve are four SU systems which share the same wireless medium. Alice is a transmitter, and intends to transmit messages to Bob and Charlie via the wireless medium. Suppose Alice and Bob have agreed on a keyed authentication scheme (implemented at the PHY layer) that allows Bob (a.k.a. "aware receiver") to authenticate the waveforms he receives from Alice. To enable authentication, Alice embeds an authentication signal into the message signal. In this model, Bob represents a regular receiver that intends to authenticate Alice's message signal. Bob can also represent a regulatory authority (e.g., FCC) that needs to ensure that Alice complies with the established spectrum rules. Charlie (a.k.a. "unaware receiver") does not know the authentication scheme and cannot authenticate Alice's waveforms at the PHY-layer, but should be able to demodulate and decode the message signal that can be authenticated at upper layers. Eve, the adversary, has knowledge of the authentication scheme but does not know the key, and hence cannot forge Alice's authentication

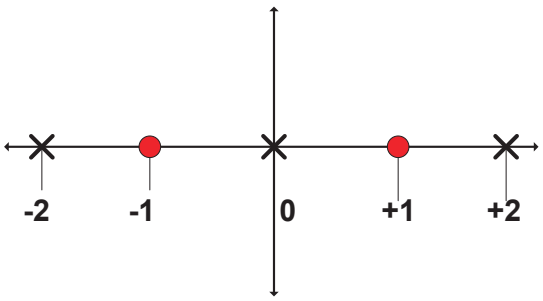


Figure 6.2: Constellation of P-DS (red circles represent the bipolar signal, and black crosses represent the duobinary signal)

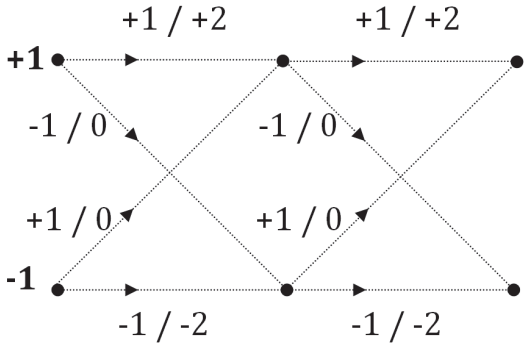


Figure 6.3: Trellis used by MLSD for P-DS

signal.

6.2 Background

Assume that $\{d_n\}$, $n = 1, 2 \cdots N$, denotes a message sequence of bits representing the message signal that needs to be transmitted, where N represents the size of the block of the message signal. Using the non-return-to-zero (NRZ) encoding, a bipolar sequence, $\{w_n\}$, is generated from the message sequence, $\{d_n\}$. Further, a duobinary symbol, y_n, is generated by adding the delayed pulse of w_n to itself. Hence, the duobinary symbol is represented by

$$y_n = w_n + w_{n-1}. \tag{6.1}$$

This equation signifies that the duobinary symbol is generated by adding a given bipolar symbol to the immediately previous bipolar symbol. If $w_n = \pm 1$, this results in a three-level output—i.e., y_n has one of three possible values: $+2$, 0 or -2 (see Figure 6.2).

Here, the duobinary symbol, y_n, can be 0 for two cases—when $w_{n-1} = +1$ is followed by $w_n = -1$, and when $w_{n-1} = -1$ is followed by $w_n = +1$. Therefore, if the receiver decodes w_{n-1} incorrectly, it affects the decoding of y_n and consequently, the detection of w_n is also

likely to be in error. This error propagation can be avoided by *precoding* the message sequence at the transmitter, i.e., the message sequence is precoded to produce a new sequence called the precoded sequence. Therefore, we refer to this signaling technique as *Precoded Duobinary Signaling* (P-DS).

In P-DS, the precoded sequence, $\{p_n\}$, for the message sequence, $\{d_n\}$, is generated using the relation $p_n = d_n \oplus p_{n-1}$, where $\oplus$ represents modulo-2 addition. Further, the bipolar sequence, $\{w_n\}$, is generated from the precoded sequence, $\{p_n\}$, using NRZ encoding. Each symbol in the duobinary sequence, $\{y_n\}$, is generated using the equation (6.1), and transmitted after RF processing. We note that the first precoded bit is generated as $p_1 = d_1 \oplus p_0$. Also, we observe that the duobinary symbol, y_1, corresponding to the bipolar symbol, w_1, is given by $y_1 = w_1 + w_0$, where w_0 and w_1 are the bipolar symbols corresponding to the precoded bits, p_0 and p_1, respectively. Hence, in P-DS, we require an extra bipolar symbol, w_0 and a corresponding precoded bit, p_0, to start the encoding of the message sequence, $\{d_n\},\ n = 1, 2 \cdots N$. The bit, p_0, is called an *initialization bit* which is usually given the value of 0. Correspondingly, the symbol, w_0, is called an *initialization state* which is usually given the value of -1.

At the receiver, after RF processing, the received signal is estimated as the duobinary sequence, $\{\hat{y}_n\}$ in the baseband. Henceforth, two decoding methods can be utilized—symbol-by-symbol detection (SSD) and maximum likelihood sequence detection (MLSD).

Using SSD method, the estimated message sequence, $\{\hat{d}_n\}$, is obtained using the following decoding decision rule.

$$\hat{d}_n = \begin{cases} 0, & \text{if } \hat{y}_n = +2 \text{ or } -2; \\ 1, & \text{if } \hat{y}_n = 0. \end{cases} \tag{6.2}$$

The bit error rate (BER) of the message signal decoded using SSD [101] is given by

$$P_{SS} = \frac{3}{4} \operatorname{erfc}\left(\frac{\pi}{4}\sqrt{\frac{E_b}{N_0}}\right), \tag{6.3}$$

where erfc, E_b and N_0 represent the complementary error function, the average bit energy,

and noise power spectral density, respectively.

Since the three-level duobinary signaling incurs an increase in the number of constellation points in Euclidean space compared to binary signaling, duobinary signaling's error performance against noise is inferior to that of binary signaling when SSD is utilized. However, the duobinary sequence in P-DS is generated from a bipolar sequence and has memory of length 1—i.e., the current state is related only to the previous state. Hence, we can use the MLSD (based on Viterbi trellis decoding) with two states (i.e., $+1$ and -1) to obtain an estimate of the transmitted bipolar sequence, $\{\hat{w}_n\}$. Figure 6.3 shows the trellis used by the MLSD, and it is generated by considering all possible transitions from each of the states. For example, an arrow from state $+1$ with the label $+1/+2$ represents a transition to the next state indicated by the left number, $+1$. The right number, $+2$, denotes the resultant signal level.

The received bipolar sequence, $\{\hat{w}_n\}$, is estimated from $\{\hat{y}_n\}$ using MLSD. Further, the estimated precoded sequence, $\{\hat{p}_n\}$, is generated from $\{\hat{w}_n\}$ using NRZ decoding. Finally, to obtain the estimated message sequence, $\{\hat{d}_n\}$, the decoding of the estimated precoded sequence is carried out as $\hat{d}_n = \hat{p}_n \oplus \hat{p}_{n-1}$, where $\oplus$ represents modulo-2 addition. The BER of the message signal using MLSD [102] is upper bounded by

$$P_{ML} = \text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right). \tag{6.4}$$

Table 6.1 provides an example illustrating the results of using P-DS encoding for the message sequence, $\{010110\}$ with the initialization bit, $p_0 = 0$. Table 6.2 illustrates the SSD of the message signal encoded in Table 6.1.

6.3 Details of P-DSA

In the following discussions, we describe our proposed scheme, viz., *P-DS for Authentication* (P-DSA). In P-DS, a known initialization bit is needed to start the encoding of the message

Table 6.1: An example illustrating P-DS encoding.

d_n		0	1	0	1	1	0
p_n	0	0	1	1	0	1	1
w_n	-1	-1	$+1$	$+1$	-1	$+1$	$+1$
y_n		-2	0	$+2$	0	0	$+2$

Table 6.2: An example illustrating SSD in P-DS.

$\hat{y}_n$	-2	0	$+2$	0	0	$+2$
$\hat{d}_n$	0	1	0	1	1	0

signal. However, we note that this initialization bit can be varied while encoding, with minimal effect on the performance of the message signal's decoding procedure. The core idea of P-DSA is to generate the embedded signal for each block of the message signal (MS_a) in such a way that the initialization bit is varied based on the authentication signal (AS_a), i.e., P-DSA uses this initialization bit as an authentication bit.

6.3.1 Embedding of AS_a into MS_a

We assume that MS_a contains blocks of binary message sequences of length N represented by $\{d_n\},\ n = 1, 2 \cdots N$. We also assume that AS_a is a binary sequence of length K generated using the scheme described in Chapter 4, and represented by $\{a_k\},\ k = 1, 2 \cdots K$. The encoding procedure of P-DSA is the same as the one for P-DS except that the precoding of each block of the message sequence is initiated using an authentication bit to be embedded. For each block of message sequence of MS_a, $\{d_n\}$, we generate the precoded sequence, $\{p_n\}$. Next, we generate the bipolar sequence $\{w_n\}$ from $\{p_n\}$ using NRZ encoding. Finally, the duobinary sequence, $\{y_n\}$, is generated from $\{w_n\}$ using equation (6.1).

As noted earlier, an initialization bit, p_0, is required to initiate the precoding of $\{d_n\}$ in each

Table 6.3: An example illustrating P-DSA encoding (the underlined bits are the authentication bits to be embedded).

d_n		0	1	0		0	1	0
p_n	$\underline{0}$	0	1	1	$\underline{1}$	1	0	0
w_n	-1	-1	$+1$	$+1$	$+1$	$+1$	-1	-1
y_n		-2	0	$+2$		$+2$	0	-2

(a) (b)

Figure 6.4: (a) MLSD for P-DS, and (b) Modified MLSD for P-DSA (The bold lines represent the possible paths emanating from the initialization state).

block. In P-DS, it is achieved by choosing a standard value for p_0. For different blocks, the same p_0 and hence the same w_0 is repeatedly used to initiate the encoding. The core idea of P-DSA is to replace the bit, p_0, in each block of MS_a with a bit from AS_a, a_k. Hence, the bipolar symbol, w_0, for the k^{th} block is generated from the authentication bit, a_k, using NRZ encoding. In the k^{th} block of the embedded signal, the first precoded bit, p_1, is generated by using the first message bit, d_1 and an authentication bit, a_k. As a result, for $a_k = 0$, the resultant precoded bit, p_1, is 0 and 1 for $d_1 = 0$ and $d_1 = 1$, respectively. Similarly, for $a_k = 1$, the resultant precoded bit, p_1, is 1 and 0 for $d_1 = 0$ and $d_1 = 1$, respectively. Table 6.3 illustrates an example of P-DSA encoding.

6.3.2 Extraction of $\widehat{MS}_b$ and $\widehat{AS}_b$

In P-DSA, we generate the embedded signal by changing the encoding procedure and accordingly change decoding procedure to extract the message signal ($\widehat{MS}_b$) and the authentication signal ($\widehat{AS}_b$) from the received signal. Note that P-DSA modifies neither the symbol mapping nor the correlation among the symbols being transmitted. Hence, the SSD is not affected by this change in encoding, while MLSD needs only a slight modification as described below.

In P-DS, at the transmitter, the precoding of each block of N bits of the message signal is started with the pre-decided initialization bit. At the receiver, MLSD starts with the initialization state which is generated from the same initialization bit. The MLSD decides on the sequence of states that is closest to the received signal in terms of Euclidean distance over the whole trellis. The complexity of this problem is significantly reduced by using the Viterbi algorithm which makes a decision on the possible paths reaching each possible state independent of other states [103]. In this case, the MLSD starts with the two paths from the initialization state to the *possible first states* corresponding to the first received duobinary symbol as shown in Figure 6.4a. Recall that trellis decoding makes a decision on the path reaching a particular state only if there are two or more paths reaching it. Hence, in P-DS, no decision is needed to select the path on each of the possible first states from the initialization state.

In P-DSA, the initialization bit is an authentication bit, and hence it also has to be estimated by the MLSD in order to decode the sequence. Hence, we need to account for the paths emanating from both the possible initialization states as shown in Figure 6.4b. Out of the two possible paths reaching each of the possible first states, we find the one that pertains to the closest first received duobinary symbol. In effect, the receiver performs SSD to determine the first symbol—i.e., it selects the closest signal level among $+2$, 0 and -2, and uses this knowledge to estimate the path from the initialization state to the state corresponding to the first symbol. Note that the signal level of $+2$ (-2) can be detected for the first zero-valued message bit if the authentication bit's state is $+1$ (-1). With the first received signal level

as 0, if the first message bit's state is +1, the authentication bit's state has to be −1 and vice versa.

6.4 Analysis

Figure 6.5 shows BER vs. E_b/N_0 curves for MS_a and AS_a when P-DSA is applied to a quadrature phase-shift keying (QPSK) modulated signal, and one authentication bit is embedded into each block of messages bits of length, $N = 16$. For comparison, we also show error performance of MS_a when P-DS is applied to a QPSK modulated signal, and no authentication signal is embedded. We observe that the performance of P-DS is very close to that of standard QPSK (without any ISI) which is used as the benchmark. This signifies that despite the addition of the ISI in the P-DS waveform, its message signal can be detected with nearly the same error performance as that of QPSK if MLSD, with sufficiently long block length, N, is used at the receiver.

We note that the error performance of MS_a in P-DSA is inferior to that of P-DS. There are two reasons for this degradation. Firstly, in P-DS, the receiver has perfect knowledge of the initialization bit's state; whereas in P-DSA, the initialization bit of each block are the authentication bits, and hence they need to be estimated. Secondly, in P-DSA, Bob employs SSD for detecting the state of the authentication bit and the first message bit of each block, but employs MLSD for rest of the message bits. Hence, the overall detection performance of MS_a in P-DSA is inferior to that of P-DS, which uses MLSD for *all* the bits in a block. As a result, the BER of the message signal in P-DSA can be upper bounded by

$$P_{MS}^{P-DSA} = \frac{1}{N} \cdot P_{SS} + \left(1 - \frac{1}{N}\right) \cdot P_{ML}, \tag{6.5}$$

where P_{SS} and P_{ML} are obtained using equations (6.3) and (6.4), respectively.

In P-DSA, the state of the authentication signal is determined by each block's first received signal level which, in turn, is estimated through comparison to the three signal levels: +2,

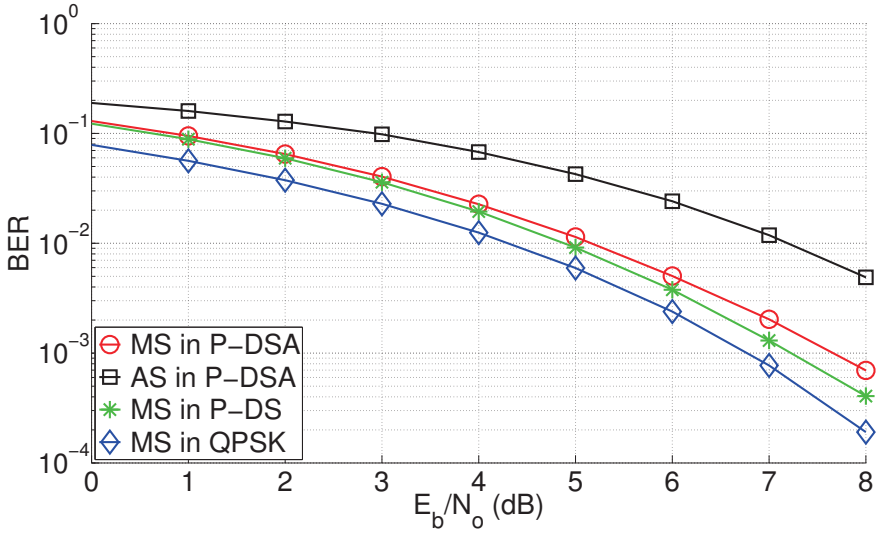
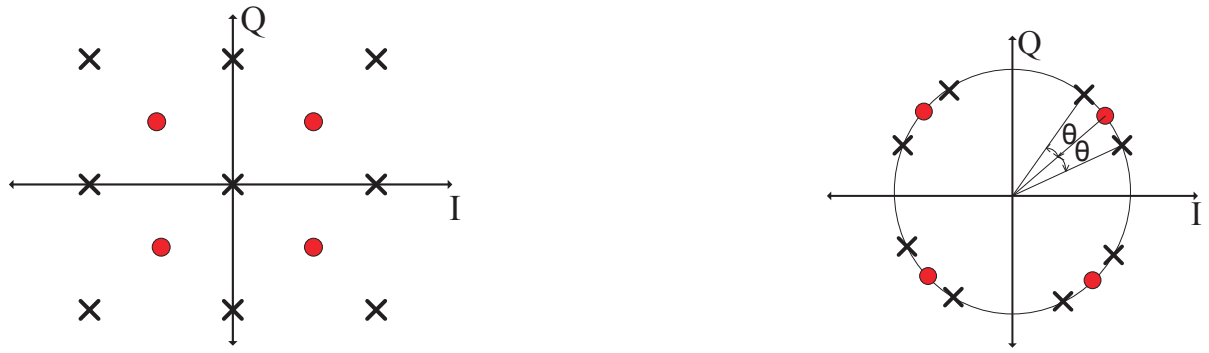


Figure 6.5: BER performance of message and authentication signals in P-DSA.

0, and -2. In essence, decoding the authentication signal depends on the performance of SSD, and does not benefit from MLSD as shown in Figure 6.5. Hence, BER of AS_a is given by P_{SS} which is calculated using equation (6.3).

6.5 Performance Evaluation

In this section, based on the performance criteria of the PHY-layer authentication schemes discussed in Section 3.5, we compare P-DSA against a benchmark scheme that is representative of the prior art: *Authentication Tagging using Modulation* (ATM) [1]. We apply P-DSA and ATM, respectively, on a QPSK modulated message signal to obtain the embedded authenticated signal. In P-DSA, the controlled ISI added to the QPSK signal results in a constellation with nine possible symbol positions as shown in Figure 6.6a. On the other hand, ATM utilizes the phase based hierarchical modulation to embed the authentication signal which leads to a constellation of eight possible symbol positions as shown in Figure 6.6b. In ATM, an authentication bit of 1 is embedded by shifting the phase of a QPSK message constellation symbol towards the Q-axis (representing quadrature-phase) by θ. An



(a) QPSK with P-DSA.

(b) QPSK with ATM.

Figure 6.6: Constellation of QPSK with P-DSA and ATM (red circles represent the message signal and black crosses represent the embedded signal).

authentication bit of 0 is embedded by shifting the phase towards the I-axis (representing in-phase) by θ. Note that both the schemes, P-DSA and ATM, do not support authentication of concurrent transmission, and blind authentication.

6.5.1 Overhead

Embedding the authentication signal in the message signal requires applying changes to the message signal itself, and thus incurs some PHY-layer overhead. For instance, the mechanism proposed in [31] results in drop in the message throughput. By design, P-DSA as well as ATM does not change the message throughput. Also, the overall average transmission power is unchanged from standard QPSK. In terms of the transmitter's and the aware receiver's complexity, ATM is advantageous compared to P-DSA. To implement ATM, the transmitter (Alice) and the receiver (Bob) only need to modify how the embedded signal is mapped to the constellation symbols. However, implementation of P-DSA is more complex—Alice needs to add controlled ISI, and Bob requires a MLSD to extract the message and the authentication signals.

6.5.2 Compatibility

This criterion dictates that a PHY-layer authentication scheme should embed the authentication signal into the message signal such that it enables the aware receiver (Bob) to extract the authentication signal, while at the same time, enables the unaware receiver (Charlie) to recover the message signal *without* requiring to change its demodulation or decoding procedure. In P-DSA, to avoid error propagation, we use precoding at the transmitter and remove the precoding to estimate the message signal at the aware receiver. Also, the embedded signal transmitted by Alice contains zero-valued signal levels as shown in Figure 6.6a. Therefore, the unaware receiver must have the knowledge of P-DSA for extracting the message signal. In contrast, in ATM, the unaware receiver does not need to change the demodulation/decoding procedure to recover the message signal—i.e., he simply treats the embedded signal as a regular QPSK modulated signal and the embedded authentication signal as noise. Therefore, ATM has the advantage over P-DSA in terms of compatibility. However, in ATM, Bob, the aware receiver, with knowledge of the embedding scheme, does no better than the unaware receiver in terms of error performance of the message signal.

6.5.3 Message Signal's Error Performance

This criterion refers to the achievable error performance (in terms of BER) when decoding the received message signal, $\widehat{MS}_b$. Figure 6.7 shows the error performance of MS_a in P-DSA with $N = 16$, ATM with phase shift of $\theta = \pi/12$ rad and ATM with phase shift of $\theta = \pi/6$ rad. In ATM, the message signal's constellation points are intentionally positioned in non-optimal positions so that the authentication signal's constellation can be superimposed on top of the message signal's constellation. Hence, as the presence of the authentication signal becomes more dominant (by increasing θ) in ATM, the BER performance of the message signal detection degrades as shown in Figure 6.7. Our scheme, P-DSA, is not constrained by such a tradeoff, and this attribute provides an important advantage in terms of error performance.

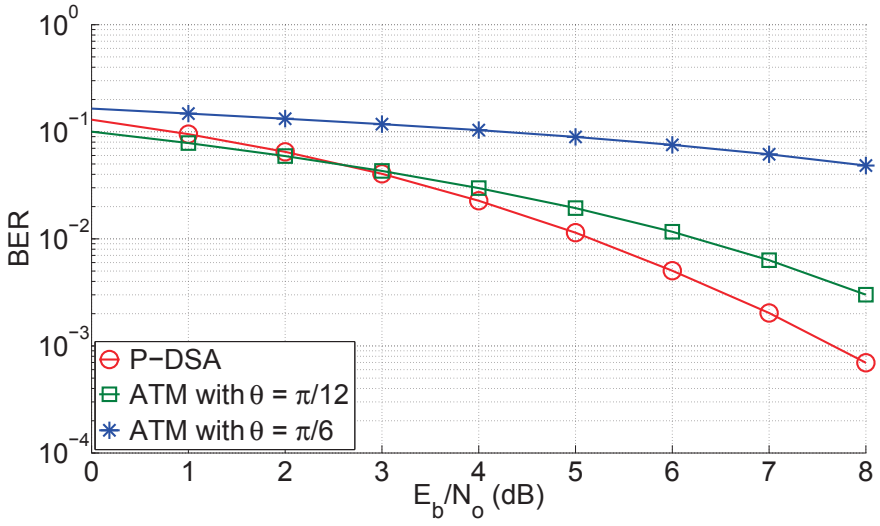


Figure 6.7: Comparison between the BER performance of the message signal in P-DSA and that in ATM.

6.5.4 Authentication Signal's Error Performance

This criterion refers to the achievable error performance when decoding the received authentication signal, $\widehat{AS}_b$. Figure 6.8 shows the error performance of AS_a in P-DSA with $N = 16$, ATM with phase shift of $\theta = \pi/12$ rad and ATM with phase shift of $\theta = \pi/6$ rad. In ATM, the message and authentication signals are embodied in two different constellations (i.e., message signal is carried in the low-resolution constellation and authentication signal is carried in the high-resolution constellation). The effect of this multi-resolution modulation can be observed when we compare ATM's curves in Figure 6.7 and 6.8. Comparing the curve of ATM with phase shift of $\theta = \pi/12$ rad in Figure 6.7 with that of Figure 6.8, we see that the BER performance of message signal is noticeably better than that of authentication signal. Moreover, we also observe that the exact opposite is true for ATM with $\theta = \pi/6$ rad. When the phase shift is $\theta = \pi/12$ rad, the shift in the constellation points (from their conventional QPSK positions) is not significant enough to cause a significant drop in BER of message signal detection. However, this relatively small shift in phase makes decoding of the authentication signal difficult, because it is carried in a high-resolution constellation.

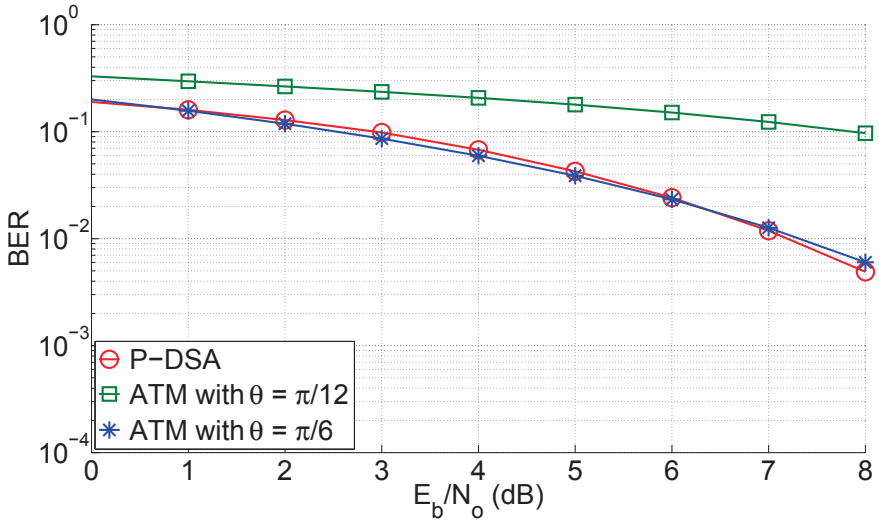


Figure 6.8: Comparison between the BER performance of the authentication signal in P-DSA and that in ATM.

When $\theta = \pi/6$ rad, the situation is reversed. From Figure 6.8, we observe that with $\theta = \pi/6$ rad, ATM has comparable BER performance compared to P-DSA for the detection of the authentication signal. However, Figure 6.7 shows that P-DSA has a significant advantage in terms of BER performance of message signal detection. On the other hand, when ATM with $\theta = \pi/12$ is used, the BER performance of message signal detection is improved (compared to ATM with $\theta = \pi/6$). However, changing from $\theta = \pi/6$ to $\theta = \pi/12$ causes a significant increase in BER for the detection of the authentication signal as shown in Figure 6.8.

For a PHY-layer authentication scheme to be viable, Bob must be able to decode both the message and the authentication signals with sufficiently good BER. ATM makes a tradeoff between the message signal's SNR and the authentication signal's SNR under the assumption of constant average power. This implies that one cannot improve the former without sacrificing the latter, and vice versa. This attribute is a fundamental drawback of blind signal superposition. P-DSA does not make the aforementioned tradeoff, and instead embeds the authentication signal by exploiting the inherent redundancy in the waveform shaping process. The resulting nine-level signal does increase the number of constellation points (thereby decreasing the minimum Euclidean distance between constellation points), but nevertheless

manages to outperform ATM in terms of BER performance. From Figures 6.7 and 6.8, and the above discussions, we can conclude that P-DSA enjoys a significant advantage over ATM.

6.5.5 Authentication Rate

In P-DSA, one bit of AS_a is transmitted in each block (of length N bits) of MS_a, which leads to an authentication rate of $1/N$. Although the authentication rate in P-DSA can be varied by changing N, decreasing N leads to a lower trellis length for MLSD. This leads to lower error performance for MS_a, which is inferred using equation (6.5). However, changing N does not affect the error performance of AS_a in P-DSA as the detection of AS_a depends only on the detection of the first received signal level in each block. On the other hand, ATM achieves an authentication rate of 1/2—one authentication bit can be inserted for every two message bits or one QPSK modulated symbol.

6.5.6 Security

This criterion determines the robustness of a PHY-layer authentication scheme against the attack carried out by Eve on the embedding and the extraction process of the authentication signal. In a PHY-layer authentication scheme, when the embedded signal with authentication signal embedded into the message signal is transmitted at the PHY-layer, Eve can launch a particular type of jamming attack specifically against the authentication signal. Hence, we propose the idea of *obstruction of authentication* (OOA) jamming attack. The OOA jamming is different from a conventional (or indiscriminate) jamming attack. The objective of conventional jamming is to prevent a targeted receiver from correctly decoding the transmitted message by generating interference of sufficient power. In contrast, the objective of OOA jamming is to generate just enough interference to prevent Bob from verifying the authenticity of the message, yet still enable him to correctly decode the message itself. OOA jamming is difficult to detect because it can readily be mistaken for naturally-occurring

noise or non-malicious interference. In certain scenarios, this may encourage Bob to treat the received message as a legitimate message without actually authenticating it. Hence, this has obvious security implications. The effectiveness of OOA jamming is dictated by the PHY-layer scheme that is used to embed the authentication signal into the message signal, and *not* the contents of the authentication signal.

Because the two schemes—viz, P-DSA and ATM—dictate the methodology by which the authentication signal is embedded into the waveform, we focus our discussions on the two schemes' resilience against OOA jamming attack that may be launched by Eve. OOA jamming can be quite effective against blind signal superposition schemes [29, 1, 104], which allocate different amounts of transmission power to the message and authentication signals respectively, including ATM. In these schemes, the message signal is embodied by a high-power constellation while the authentication signal is carried on a low-power constellation. In this case, Eve can emit just enough interference to exploit the power difference, and thus prevent decoding of the authentication signal but enable decoding of the message signal. However, in P-DSA, to obstruct Bob from decoding the authentication signal, Eve would need to generate interference that is sufficiently powerful to also make decoding of the message signal impossible. Hence, P-DSA is robust to OOA jamming.

6.6 Experimental Validation

We implemented P-DSA as a prototype using two Universal Software Radio Peripheral (USRP) radios, one each for Alice (transmitter) and Bob (aware receiver). We used National Instruments' LabVIEW as the system-design platform to configure the USRPs. Alice and Bob use the PHY-layer protocol discussed in IEEE 802.11af draft standard [105] to communicate with each other at 900 MHz. Alice generates the message signal using orthogonal frequency division multiplexing (OFDM). Alice also embeds an authentication signal into its message signal using P-DSA so that Bob is able to authenticate Alice.

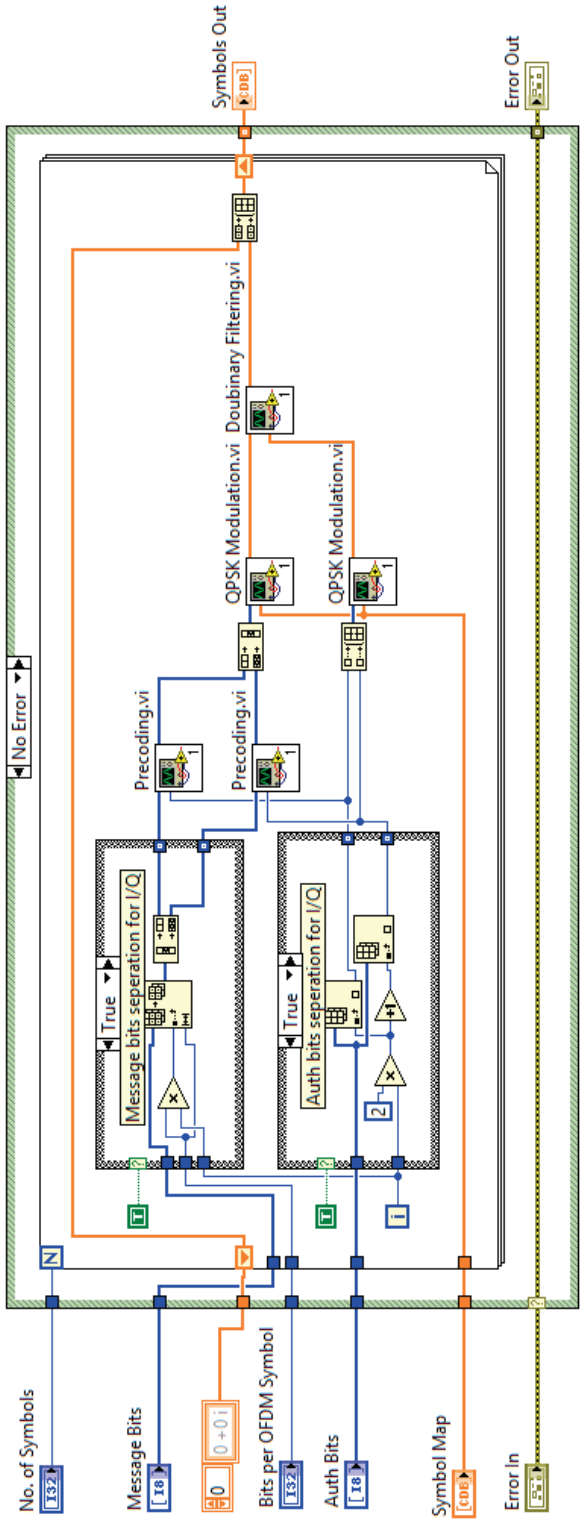


Figure 6.9: LabVIEW VI illustrating the implementation of P-DSA.

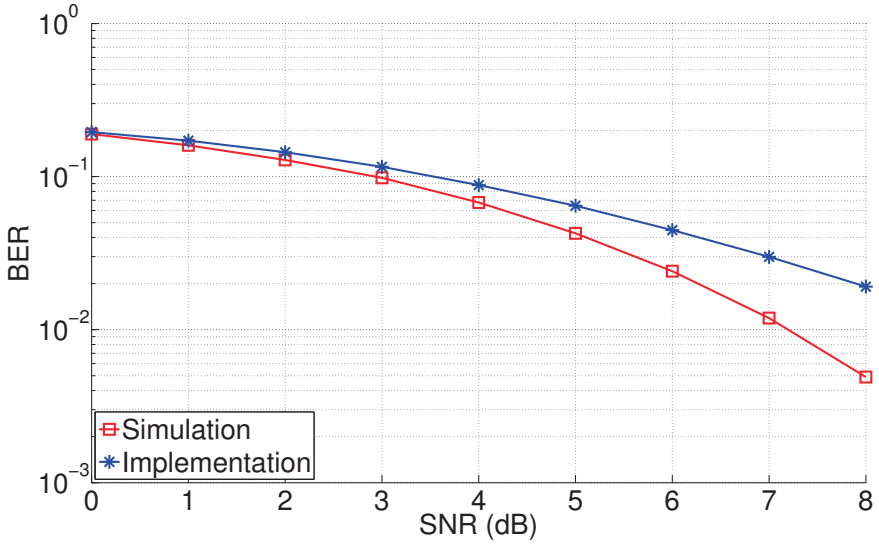


Figure 6.10: BER performance of the authentication signal in the LabVIEW implementation of P-DSA.

6.6.1 Model and Assumptions

The two radios are placed approximately 1 meter away from each other in an indoor environment. The distance between the radios is limited by the fact that both the radios need to be connected to the computer running the LabVIEW application through network cables. Hence, to obtain variable SNR values (from 0 dB to 8 dB), we add Gaussian noise at Bob in addition to the channel-induced and thermal noise added to the signal transmitted over-the-air.

6.6.2 Design

We utilize QPSK as the modulation scheme for the message signal. The data contained in the message signal consists of a time-stamp and message text, and it is transmitted without any error correction coding. The authentication signal also consists of a time-stamp and authentication text without any error correction coding. The authentication

signal is embedded into the message signal using P-DSA with a block length of $N = 16$. Bob demodulates and decodes the received signal, and then separates the message and authentication signals. The received message and authentication signals are synchronized using the time-stamp, and compared with the transmitted message and authentication signals to calculate their BER, respectively.

Figure 6.9 shows the LabVIEW VI of the steps needed to embed an authentication symbol into the message signal using P-DSA. The message bits and the authentication bits are separated into in-phase and quadrature-phase streams. After precoding the message bits in each block with an authentication bit, the precoded sequence is mapped to QPSK symbols. Finally, duobinary signaling is carried out for each block. Further, conventional processes like performing inverse fast fourier transform (IFFT), adding cyclic prefix, and adding preamble symbols are performed to generate the OFDM signals to be transmitted over-the-air.

6.6.3 Results

Figure 6.10 shows the BER performance of the authentication signal (AS_a) for the LabVIEW implementation of P-DSA. As a benchmark, a BER performance curve generated from Matlab simulations using the same PHY-layer parameters is also presented. We observe that the BER performance of the LabVIEW implementation is slightly inferior to that of the simulations. This phenomenon can be attributed to the fact that the channel noise is Gaussian in the simulations, whereas the channel noise is not truly Gaussian in the over-the-air experiments.

6.7 Summary

We proposed a novel PHY-layer authentication scheme referred to as *Precoded Duobinary Signaling for Authentication* (P-DSA). P-DSA is fundamentally different from the prior art, and

it is not constrained by the tradeoff that constrains the blind signal superposition schemes. Although P-DSA increases the number of points in the signal constellation (compared to conventional binary signaling), our simulation results show that it achieves improved error performance of message and authentication signals compared to the prior art without sacrificing message throughput or requiring an increase in transmission power. P-DSA inherits such desirable attributes at the cost of increased transmitter/receiver complexity.

Chapter 7

FEAT: Frequency Offset Embedding for Authenticating Transmitters

In this chapter, we propose a novel blind transmitter authentication (BTA) scheme (discussed in Section 1.2.2) called *Frequency offset Embedding for Authenticating Transmitters* (FEAT). To the best of our knowledge, FEAT is the first scheme that satisfies the three requirements of an ideal BTA scheme. FEAT modifies the frequency offset of each frame of the message signal to embed the authentication signal into the message signal. This is achieved in such a way that the authentication signal does not interfere with the decoding process of the message signal. Also, the authentication signal can be estimated at the blind receiver with only limited knowledge about the transmission parameters by estimating the frequency offset of each frame. We show that FEAT outperforms the existing PHY-layer authentication approaches in all of the performance criteria that were considered. We evaluate FEAT using simulation results and theoretical analysis. In addition, we verify the validity of FEAT by carrying out experiments with an actual implementation.

The rest of the chapter is organized as follows. We provide the model and assumptions in Section 7.1. We discuss FEAT in Section 7.2, and analyze it in Section 7.3. We evaluate FEAT by comparing with the prior art in Section 7.4. We discuss a prototype implemen-

tation of FEAT in Section 7.5. Section 7.6 concludes the chapter by highlighting the main contributions.

7.1 Model and Assumptions

We assume that Alice, Bob, Charlie, Dave and Eve are five users which share the same wireless medium, as shown in Figure 7.1. Alice intends to transmit messages to Bob and Charlie via the wireless medium as per the rules established for dynamic spectrum sharing. Alice utilizes CP-OFDM for its message signals, but can reconfigure its PHY-layer parameters as per the requirements for the wireless medium. Alice conveys the information about these parameters to Bob as well as Charlie so that they can demodulate and decode Alice's message signal. Dave (a.k.a. "blind receiver") represents a regulatory entity that needs to authenticate Alice. Suppose Alice and Dave have agreed on a keyed authentication scheme that enables Dave to blindly authenticate the waveforms that he receives from Alice. For this to work, we must require Alice's radio to embed an authentication signal into her message signal's waveform using the agreed authentication scheme, and Dave must have the capability to extract and decode the authentication signal from the received signal. Bob (a.k.a. "aware receiver") has knowledge about the message signaling scheme and the authentication scheme. This means that Bob can decode the message signal as well as the authentication signal from the received waveforms. Charlie (a.k.a. "unaware receiver") does not know the authentication scheme and cannot authenticate Alice's waveforms, but should be able to demodulate and decode Alice's message signal. Eve represents an adversary, and she is able to launch various types of attacks against Alice and Bob, e.g., jamming attacks.

Further, we assume that Dave receives signals from Alice and Eve with low SINR and significant multipath. Also, there may be simultaneous transmissions from Alice and Eve on the same spectrum band. This means that Dave may receive signals from Alice and Eve at the same time. Hence, Dave should be able to authenticate even when the SINR is below

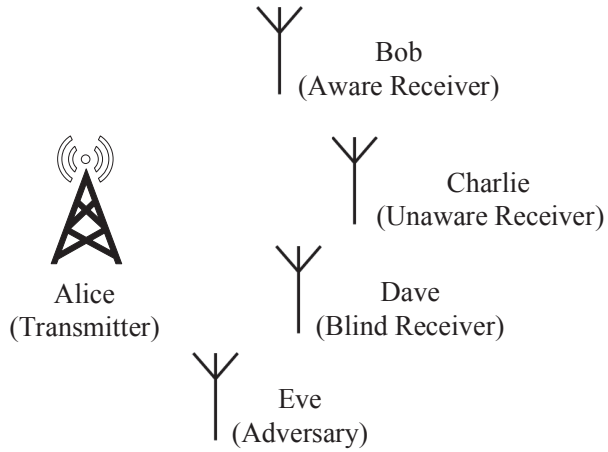


Figure 7.1: Authentication scenario in BTA.

0 dB. Usually, it is very difficult for a receiver to decode the message signal under such harsh channel conditions [106]. It is also assumed that Dave is aware of the fact that CP-OFDM is employed by Alice and Eve to modulate the message signals. Dave also knows the center frequency and the sampling frequency of their signals; these parameters are typically standardized as part of an air-interface standard [105].

7.2 Details of FEAT

We propose a BTA scheme that we refer to as *F*requency offset *E*mbedding for *A*uthenticating *T*ransmitters (FEAT). In the following text, we describe FEAT in the context of the authentication scenario depicted in Figure 7.1.

Alice embeds the authentication signal in the form of embedded frequency offset (EFO) in each frame of the message signal in the baseband. The embedded signal in the baseband is sent to the oscillator where it gets up-converted and transmitted along with the inherent carrier frequency offset (CFO) due to the inaccurate oscillator. This overall frequency offset does not affect the decoding procedure of the message signal by Bob and Charlie as being the intended receivers, they estimate and correct any frequency offset present in the received

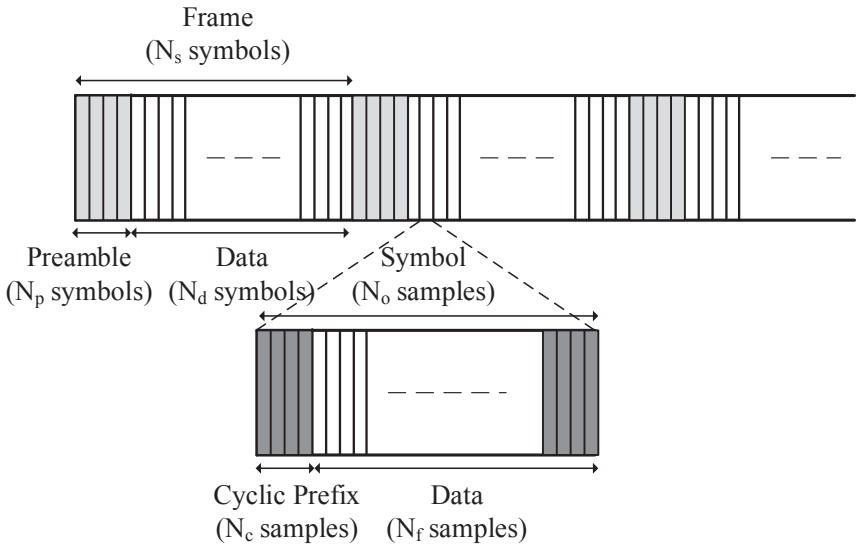


Figure 7.2: Structure of the OFDM message signal.

signal with the help of the preamble symbols, and the pilot samples. Dave estimates the authentication signal blindly. Further, we discuss in detail the generation of the message signal (MS_a) and authentication signal (AS_a), and embedding of AS_a into MS_a by Alice followed by extraction of the message signal ($\widehat{MS}_b$) and authentication signal ($\widehat{AS}_b$) by Bob, extraction of the message signal (MS_c) by Charlie, extraction of the authentication signal ($\widehat{AS}_d$) by Dave, and verification of $\widehat{AS}_b$ and $\widehat{AS}_d$.

7.2.1 Generation of MS_a

The message data to be transmitted is assumed to be a sequence of quadrature amplitude modulated (QAM) samples which are statistically independent and identically distributed with zero mean and average power represented by σ_s^2. For each OFDM symbol, Alice generates N_f samples by taking the Inverse Fast Fourier Transform (IFFT) of N_u QAM samples corresponding to N_u non-zero sub-carriers loaded with data or pilot samples. The last N_c samples out of the N_f samples are repeated at the beginning of the N_f samples as the cyclic prefix (CP) to generate an OFDM symbol of $N_o = N_f + N_c$ samples. The message signal is

transmitted in frames, and each frame contains $N_s = N_p + N_d$ OFDM symbols, where N_p represents the number of symbols carrying the preamble and N_d represents the symbols carrying data. The samples of a frame is represented by $\{s(n)\}$, where $n = 0, 1, \cdots N_s \cdot N_o - 1$. Figure 7.2 illustrates the symbol and frame structure of MS_a.

7.2.2 Generation of AS_a

The authentication signal AS_a of K_a bits can be generated by using the schemes proposed in Chapter 4 or Chapter 5.

7.2.3 Embedding of AS_a into MS_a

For embedding the authentication signal into the message signal, we propose a novel scheme called *F*rame *F*requency *M*odulation (FFM) where the frequency offset of each frame of the message signal is modified (modulated) according to the authentication signal. FFM of order M (M-FFM) is represented by a set of M possible frequency offsets corresponding to $M = 2^b$ possible b-bit authentication symbols. Here, an authentication symbol is defined as a set of b authentication bits, and is obtained by using b-bit Gray code. The set of frequency offsets in M-FFM can be represented by $\{f_m\}$ such that

$$f_m = f_a \cdot \left(1 - 2 \cdot \frac{m-1}{M-1}\right), \tag{7.1}$$

where $m = 1, 2, \cdots, M$, and f_a is the maximum positive frequency offset that can be used to embed the authentication signal into a frame of the message signal. Figures 7.3a and 7.3b represent the mapping schemes for 1-bit authentication symbols and 2-bit authentication symbols, respectively. Note that $f_{M-m+1} = -f_m$, for $m = 1, 2, \cdots, \frac{M}{2}$.

In k^{th} frame of the message signal, we embed the authentication symbol, represented by a_k, by embedding a frequency offset, f_k. Hence, for $n = 0, 1 \cdots N_s \cdot N_o - 1$, each sample of a frame of the embedded signal in the baseband is given by $x(n) = s(n) \cdot e^{j2\pi \frac{f_k}{F_s} n}$, where F_s is

(a) M=2 (b) M=4

Figure 7.3: Mapping of authentication symbols to frequency offsets in M-FFM

the sampling frequency. The embedded signal is up-converted to the carrier frequency (F_c) and transmitted. Assuming that CFO due to the inaccurate oscillator at Alice is f_t, the total frequency offset of the transmitted signal is $f_k + f_t$.

7.2.4 Extraction of $\widehat{MS}_b$, $\widehat{AS}_b$, and $\widehat{MS}_c$

After down-converting and sampling the received signal, Bob, with the knowledge of the preamble symbols and the pilot samples, can estimate and correct the frequency offset in each frame and extract the message signal, $\widehat{MS}_b$. Also, Bob maps the frequency offset in each of the frames to the closest one among $\{f_m\}$, for $m = 1, 2 \cdots M$, given by equation (7.1) and estimates the authentication signal, $\widehat{AS}_b$. Charlie, also equipped with the knowledge of the preamble symbols and the pilot samples, can correct the frequency offset in each frame and extract the message signal, MS_c. Since Charlie is not interested in the frequency offsets of the frames of the message signal, the information contained in these frequency offsets is simply discarded by Charlie.

7.2.5 Extraction of $\widehat{AS}_d$

Dave, the blind receiver, does not have the knowledge about the preamble symbols or the pilots samples inserted in the message signal. Hence, to blindly estimate the transmitted authentication signal, Dave needs to carry out four tasks—signal detection and sampling, symbol synchronization, frame synchronization, and frame frequency estimation.

Signal Detection and Sampling

Since Dave has the knowledge of the center frequency and sampling frequency of the transmitted signal, it down-converts and samples the received signal in the considered frequency band. Assuming a Gaussian channel, Dave observes the received signal with N_r discreet samples which can be represented by

$$r(n) = e^{j2\pi\frac{f_r}{F_s}n} \cdot x(n) + w(n), \text{ for } n = 0, 1, \cdots N_r - 1,$$

where f_r is the frequency offset due to the oscillator at Dave. The additive noise $w(n)$ is assumed to be independent of $x(n)$, and circularly complex Gaussian with zero mean, and σ_w^2 variance. It can be observed that the frequency offset in each frame of the received signal has one constant part, $f_c = f_t + f_r$, and a variable part, f_k, carrying authentication signal.

Symbol Synchronization

Estimation of symbol boundaries includes estimation of the IFFT size ($\widehat{N}_f$), the CP size ($\widehat{N}_c$) and the sample offset ($\widehat{\alpha}$). Dave estimates these three parameters using the sub-optimal maximum likelihood (ML) scheme described in [107] which utilizes the correlation in the received signal induced due to CP. The likelihood function, $\Lambda(\mathbf{r}, \widetilde{N}_f, \widetilde{N}_c, \widetilde{\alpha})$, can be expressed by

$$\Lambda = \frac{1}{\widetilde{N}_n \cdot \widetilde{N}_c} \sum_{i=0}^{\widetilde{N}_n - 1} \sum_{l=0}^{\widetilde{N}_c - 1} r^* \left(i \cdot (\widetilde{N}_f + \widetilde{N}_c) + \widetilde{\alpha} + l \right) \cdot r \left(i \cdot (\widetilde{N}_f + \widetilde{N}_c) + \widetilde{N}_f + \widetilde{\alpha} + l \right).$$

where $r^*(n)$ is the complex conjugate of $r(n)$, and $\widetilde{N}_n = \lfloor (N_r - \widetilde{\alpha})/(\widetilde{N}_f + \widetilde{N}_c) \rfloor$. Here, $\lfloor v \rfloor$ denotes the largest integer less than or equal to v. $\widehat{N}_f$, $\widehat{N}_c$ and $\widehat{\alpha}$ can be estimated as

$$\widehat{N}_f, \widehat{N}_c, \widehat{\alpha} = \underset{\widetilde{N}_f, \widetilde{N}_c, \widetilde{\alpha}}{\operatorname{argmax}} \left| \Lambda \left(\mathbf{r}, \widetilde{N}_f, \widetilde{N}_c, \widetilde{\alpha} \right) \right|.$$

where $|v|$ denotes the absolute value of v. Dave obtains the estimate of the constant part of the frequency offset as

$$\widehat{f_c} = \frac{F_s}{2\pi \widehat{N}_f} \cdot \angle\lambda, \tag{7.2}$$

where $\lambda = \Lambda(\mathbf{r}, \widehat{N}_f, \widehat{N}_c, \widehat{\alpha})$, and $\angle c$ denotes the polar angle of the complex number c. After symbol synchronization, the received signal is represented by $r_s(n) = r(\widehat{\alpha} + n)$ for $n = 0, 1, \cdots N_r - \widehat{\alpha}$. Note that the perfect symbol synchronization is achieved when $\widehat{N}_f = N_f$, $\widehat{N}_c = N_c$, and $\widehat{\alpha} = \alpha$, where α is the actual sample offset. In this case, the theoretical value of λ can be obtained as discussed below.

For very large N_r, when no authentication signal is embedded, $\lambda = \sigma_s^2 \cdot e^{j\epsilon f_c}$, where f_c is the constant frequency offset, and $\epsilon = 2\pi N_f / F_s$ [107]. When FEAT is utilized to embed the authentication signal, we could define M sets of frames with the frequency offsets $f_c + f_k$, where $f_c = f_t + f_r$, and $f_k = f_m$ for $m = 1, 2, \cdots, M$. Assuming that the authentication symbols are statistically independent and identically distributed, λ is given by

$$\begin{aligned}
\lambda &= \frac{\sigma_s^2}{M} \cdot \sum_{m=1}^{M} e^{j\epsilon(f_c + f_m)} \\
&= \frac{\sigma_s^2}{M} \cdot e^{j\epsilon f_c} \cdot \left(\sum_{m=1}^{M/2} e^{j\epsilon f_m} + \sum_{m=M/2+1}^{M} e^{j\epsilon f_m} \right) \\
&= \frac{\sigma_s^2}{M} \cdot e^{j\epsilon f_c} \cdot \left(\sum_{m=1}^{M/2} e^{j\epsilon f_m} + \sum_{m=1}^{M/2} e^{j\epsilon f_{M-m+1}} \right) \\
&= \frac{\sigma_s^2}{M} \cdot e^{j\epsilon f_c} \cdot \left(\sum_{m=1}^{M/2} e^{j\epsilon f_m} + \sum_{m=1}^{M/2} e^{-j\epsilon f_m} \right) \\
&= \frac{2\sigma_s^2}{M} \cdot e^{j\epsilon f_c} \cdot \sum_{m=1}^{M/2} \cos \epsilon f_m \\
&= \frac{2\sigma_s^2}{M} \cdot e^{j\epsilon f_c} \cdot \sum_{m=1}^{M/2} \cos \epsilon f_a \left(1 - 2 \cdot \frac{m-1}{M-1} \right). \qquad (7.3)
\end{aligned}$$

Frame Synchronization

Estimation of frame boundaries includes estimation of the total number of symbols in a frame ($\widehat{N}_s$), and the symbol offset ($\widehat{\beta}$). Dave estimates these two parameters using the correlation among the preamble symbols of the consecutive frames of the received signal. The likelihood

function, $\Psi(\boldsymbol{r_s}, \widetilde{N}_s, \widetilde{\beta})$, can be expressed by

$$\Psi = \frac{1}{\widetilde{K}_r \cdot \widehat{N}_o} \sum_{k=0}^{\widetilde{K}_r - 1} \sum_{l=0}^{\widehat{N}_o - 1} r_s^* \left(k \cdot \widetilde{N}_s \cdot \widehat{N}_o + \widetilde{\beta} \cdot \widehat{N}_o + l \right) \cdot \; r_s \left((k+1) \cdot \widetilde{N}_s \cdot \widehat{N}_o + \widetilde{\beta} \cdot \widehat{N}_o + l \right).$$

where $\widetilde{K}_r = \lfloor (N_r - \widehat{\alpha} - \widetilde{\beta} \cdot \widehat{N}_o)/(\widetilde{N}_s \cdot \widehat{N}_o) \rfloor$, and $\widehat{N}_o = \widehat{N}_f + \widehat{N}_c$. Hence, $\widehat{N}_s$, and $\widehat{\beta}$ can be estimated as

$$\widehat{N}_s, \widehat{\beta} = \underset{\widetilde{N}_s, \widetilde{\beta}}{\operatorname{argmax}} \left| \Psi \left(\boldsymbol{r_s}, \widetilde{N}_s, \widetilde{\beta} \right) \right|.$$

The number of received frames is obtained as $\widehat{K}_r = \lfloor (N_r - \widehat{\alpha} - \widehat{\beta} \cdot \widehat{N}_o)/(\widehat{N}_s \cdot \widehat{N}_o) \rfloor$. After frame synchronization, the received signal is represented by $r_f(n) = r_s(\widehat{\beta} \cdot \widehat{N}_o + n)$ for $n = 0, 1, \cdots \widehat{K}_r \cdot \widehat{N}_s \cdot \widehat{N}_o - 1$. Note that the perfect frame synchronization is achieved when $\widehat{N}_s = N_s$, and $\widehat{\beta} = \beta$, where β is the actual symbol offset. In this case, the theoretical value of $\psi = \Psi(\boldsymbol{r_s}, \widehat{N}_b, \widehat{\beta})$ can be obtained as discussed below.

For very large N_r, when no authentication signal is embedded, each frame has the same frequency offset and hence the relative frequency offset between consecutive frames is zero which means $\psi = \sigma_s^2$. When FEAT is utilized to embed the authentication signal, we could define M^2 sets of relative frequency offsets between two consecutive frames—M sets of frequency offset 0, $M-1$ sets of frequency offset $+\frac{2f_a}{M-1}$, and so on. Hence, ψ is given by

$$\begin{aligned}
\psi &= \frac{\sigma_s^2}{M^2} \cdot M + \frac{\sigma_s^2}{M^2} \sum_{m=1}^{M-1} \frac{M-m}{N_o} \sum_{l=0}^{N_o-1} e^{j2\pi \frac{2mf_a}{(M-1)F_s} l} + \frac{\sigma_s^2}{M^2} \sum_{m=1}^{M-1} \frac{M-m}{N_o} \sum_{l=0}^{N_o-1} e^{j2\pi \frac{-2mf_a}{(M-1)F_s} l} \\
&= \frac{\sigma_s^2}{M^2} \cdot \left(M + \sum_{m=1}^{M-1} \frac{M-m}{N_o} \sum_{l=0}^{N_o-1} 2\cos 2\pi \frac{2mf_a}{(M-1)F_s} l \right) \\
&\approx \frac{\sigma_s^2}{M^2} \cdot \left(M + \sum_{m=1}^{M-1} \frac{(M-m) \cdot \sin 4\pi \frac{mf_a}{(M-1)F_s} N_o}{N_o \cdot \sin 2\pi \frac{f_a}{F_s}} \right). \qquad (7.4)
\end{aligned}$$

Frame Frequency Estimation

Having synchronized with the received signal, Dave estimates the correlation between the CP samples and the corresponding data samples in the symbols of each of the frames. For

$k = 0, 1, \cdots \widehat{K}_r$, the correlation is given by

$$\Phi(k) = \frac{1}{\widehat{N}_s \cdot \widehat{N}_c} \sum_{i=0}^{\widehat{N}_s-1} \sum_{l=0}^{\widehat{N}_c-1} r_f^* \left(k \cdot \widehat{N}_s \cdot \widehat{N}_o + i \cdot \widehat{N}_o + l\right) \cdot r_f \left(k \cdot \widehat{N}_s \cdot \widehat{N}_o + i \cdot \widehat{N}_o + \widehat{N}_f + l\right).$$

Hence, the frequency offset for each frame is estimated as

$$\widehat{f}_o(k) = \frac{F_s}{2\pi \widehat{N}_f} \angle \Phi(k). \tag{7.5}$$

The estimate of frequency offset embedded in the frame through M-FFM is obtained by $\widehat{f}_k = \widehat{f}_o(k) - \widehat{f}_c$, where $\widehat{f}_c$ is obtained from equation (7.2). Finally, Dave maps $\widehat{f}_k$ to the closest one among $\{f_m\}$, for $m = 1, 2 \cdots M$, given by equation (7.1), and estimates the authentication symbol of $\widehat{AS}_d$ which is denoted as $\widehat{a}_k$.

7.2.6 Verification of $\widehat{AS}_b$ and $\widehat{AS}_d$

Having estimated the authentication signal, $\widehat{AS}_b$ or $\widehat{AS}_d$, its authenticity is verified by utilizing the techniques discussed in Chapter 4 or 5.

7.3 Analysis

In this chapter, we evaluate FEAT using Matlab-based simulation results. Specifically, we discuss the error performance of the authentication signal when Dave is the receiver. We also discuss security issues relevant to FEAT.

7.3.1 Error Performance

To analyze the error performance of the authentication signal in FEAT, we assume that perfect symbol and frame synchronization have been achieved by Dave. An error in the authentication symbol means $\widehat{a}_k \neq a_k$ which occurs when the mapping of estimated EFO,

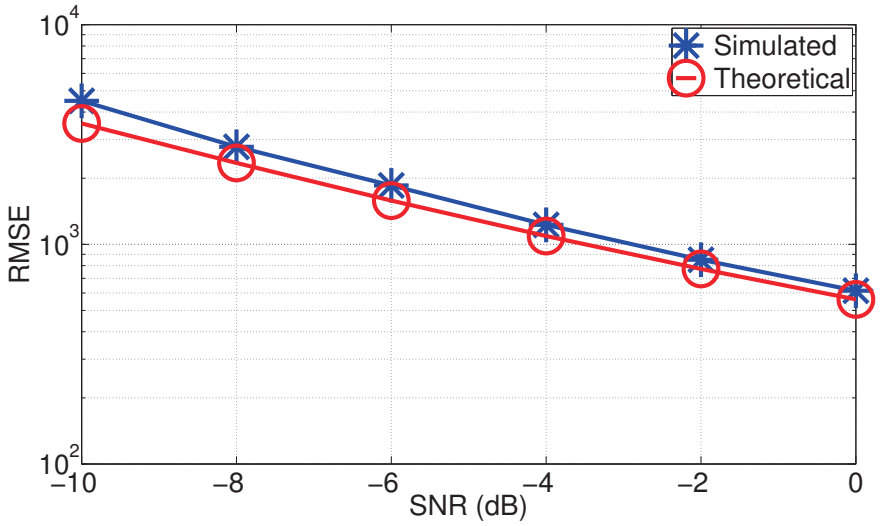


Figure 7.4: Theoretical CRLB and simulated RMSE in the estimate of $\widehat{f_k}$ at the blind receiver with $M = 2, f_a = 5$ kHz, $N_f = 64, N_c = 16, N_s = 50$.

$\widehat{f_k}$, to the closest one among $\{f_m\}$, for $m = 1, 2 \cdots M$, leads to a different EFO as compared to the transmitted EFO, f_k. This happens when the error in the estimate of the EFO exceeds the magnitude of half of the difference between two consecutive EFOs, i.e., $\left|\widehat{f_k} - f_k\right| > \frac{f_a}{M-1}$. Theoretically, the mean square error (MSE) of the estimate of $\widehat{f_k}$ is lower bounded by the Cramer-Rao Lower-Bound (CRLB) [108, 109]. We obtain the CRLB of the estimate of $\widehat{f_k}$ in FEAT as

$$CRLB = \frac{1}{8\pi^2 N_c} \cdot \left(\frac{1}{\rho^2} + \frac{2}{\rho}\right) \cdot \frac{F_s^2}{N_f^2 N_s}, \tag{7.6}$$

where $\rho = \sigma_s^2/\sigma_w^2$, represents the SNR. In Figure 7.4, we present the root mean square error (RMSE) of the estimate of $\widehat{f_k}$ at different SNRs. Note that the simulated RMSE in FEAT is quite close to its theoretical bound given by square-root of the CRLB. The RMSE vs. SNR curve helps to estimate the error performance of $\widehat{AS}_d$ at a particular SNR given the specific values of different parameters (presented in equation (7.6)). For instance, in Figure 7.4, RMSE of the estimate of $\widehat{f_k}$ at SNR of -6 dB is 2 kHz. Hence, in this example, we can estimate the error performance of $\widehat{AS}_d$ when $f_a = 5$ kHz and $M = 2$. However, since the frequency estimate $\widehat{f_k}$ is non-Gaussian in nature, we analyze the effect of different parameters

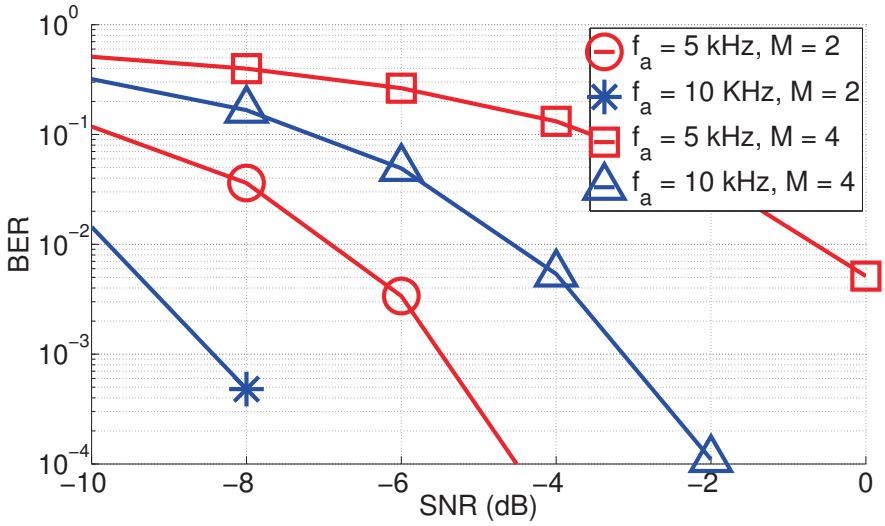


Figure 7.5: Error performance of $\widehat{AS}_d$ with $N_f = 64, N_c = 16, N_s = 50$.

on the error performance of the authentication signal through simulation where the sampling frequency F_s is chosen to be 5 MHz [105].

Effect of ρ

In Figure 7.5, when we observe the curve with $f_a = 5$ kHz and $M = 2$, we note that FEAT is quite robust against noise, and the error performance improves (i.e., BER decreases) significantly with increase in SNR, e.g., BER ≈ 0.03 at SNR $= -8$, and BER ≈ 0.003 at SNR $= -6$. This is because each frame of the message signal contains a large number of samples $(N_s \cdot N_o)$ which are used to estimate one symbol of the authentication signal.

Effect of f_a

As the largest possible value of EFO, f_a, is increased, BER of $\widehat{AS}_d$ decreases as observed in Figure 7.5. This is because by increasing EFO, we effectively account for a larger margin of error in $\widehat{f}_k$. However, there are some limitations on the value of f_a as discussed below.

In the presence of very large number of samples present for synchronization, the performance

of the proposed synchronization algorithm depends mainly on f_a. From equation (7.3), we note that the absolute value of λ decreases by increasing f_a. This means that the probability of detection of the peak of Λ decreases by increasing f_a. Again, from equation (7.4), we observe that the absolute value of ψ decreases by increasing f_a. This means that the probability of detection of the peak of Ψ decreases by increasing f_a. On the other hand, by increasing f_a, we can enhance the error performance of FEAT. Therefore, we need to find theoretical bounds on f_a.

From equation (7.3), for $M = 2$, λ is given by

$$\lambda = \sigma_s^2 \cdot e^{j\epsilon f_c} \cdot \cos \epsilon f_a.$$

where $\epsilon = 2\pi N_f / F_s$. Hence, the maximum value of λ is achieved when $f_a = 0$. This means that the synchronization is achieved with highest accuracy when no authentication signal is embedded into the message signal. On the other hand, when $f_a = \pi/(2\epsilon) = F_s/(4\pi N_f)$, the absolute value of λ is 0. This means that it is difficult to achieve the synchronization if the EFO is close to $F_s/(4\pi N_f)$. Hence, to achieve sufficient level of robustness for synchronization of the received signal and for extraction of the authentication signal, f_a needs to be sufficiently larger than 0, but sufficiently smaller than $F_s/(4\pi N_f)$.

Moreover, in the existing standards describing PHY-layer specifications, there is a limited margin allowed for the carrier frequency offset (CFO) in the message signals due to inaccurate oscillators at the transmitters and the receivers. For instance, as per IEEE 802.11g [110], the absolute value of CFO due to an inaccurate oscillator should be less than 25 ppm of the carrier frequency. Hence, we need to ensure that $f_t + f_a \leq F_o$, where F_o is the allowed frequency offset as per the standard.

Effect of M

While FEAT with $M = 2$ can carry only 1 authentication bit per frame of the message signal, but FEAT with $M = 4$ can carry 2 authentication bits per frame of the message signal. This

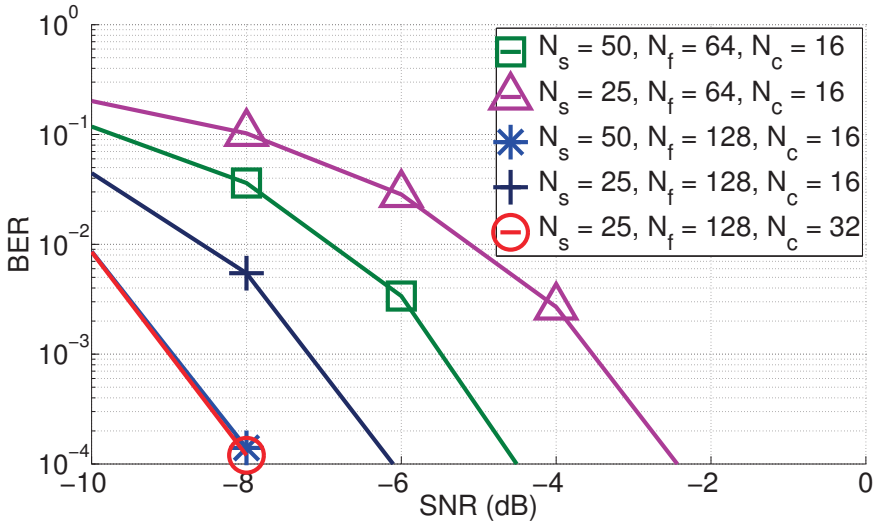


Figure 7.6: Error performance of $\widehat{AS}_d$ with $M = 2$, $f_a = 5$ kHz.

means that the authentication rate (defined in Section 3.5) is increased by increasing M. However, as shown in Figure 7.5, as M increase, the BER of $\widehat{AS}_d$ increases significantly. This means that M leads to a trade-off between the error performance and the authentication rate of $\widehat{AS}_d$. This trade-off may play an important role in the cases where the size of data to be communicated between the transmitter and the intended receivers, is small.

In order to authenticate Eve, Dave should receive all the bits from at least one complete authentication sequence. Note that in order to verify the authentication signal, at least one complete authentication sequence should be received by Dave. This means that the estimated number of frames of received signal $\widehat{K}_r$ should be greater than the length of one complete authentication sequence, K_a, i.e., $\widehat{K}_r \geq K_a$. This means that for FEAT with $M = 2$, the number of frames transmitted by Eve should be more than the length of one authentication sequence which is K_a. However, when the size of data is small, the number of frames being transmitted can be significantly small. Hence, the authentication rate needs to be increased at the cost of the error performance to ensure embedding of the authentication bits of at least one authentication sequence. This will allow the transmitter to be authenticated for all its transmission including the burst mode.

Effect of N_f and N_c

In Figure 7.6, we observe that the BER decreases by increasing N_f and N_c. Recall that N_c (CP size) is the number of samples in each OFDM symbol which are correlated with their corresponding data samples for frequency estimation. This implies that with the increase in N_c, the estimation error in frequency decreases leading to the decrease in BER of $\widehat{AS}_d$.

Effect of N_s

In Figure 7.6, we observe that increasing N_s (frame size) leads to an improvement in error performance, i.e., we achieve lower BER of $\widehat{AS}_d$. However, larger frame size also leads to lower frame rate which results into lower authentication rate. Hence, we again observe a trade-off between the authentication rate and the error performance of $\widehat{AS}_d$ in terms of N_s. We also observe that when the total number of CP samples in a frame given by $N_c \cdot N_s$ (used for correlation to estimate an authentication symbol) remains the same at a particular value of N_f, the BER of $\widehat{AS}_d$ remains the same.

Moreover, the value of N_s leads to another trade-off between the authentication rate and the transparency, which is one of the main issues that we address through FEAT. We need to use the unit for transmitting one authentication symbol as a frame since we aim to embed the authentication in an absolute transparent manner. In other word, FEAT allows for the presence of unaware receivers (those who do not know about FEAT) in the network, e.g., Charlie. However, if the network environment does not require the condition of absolute transparency (i.e., the network does not have an unaware receiver), we could embed a frequency offset in any number (as low as 1) of OFDM symbols. However, as shown in Figure 7.6, decreasing the number of symbols for frequency estimation significantly reduces the error performance. Hence, we can achieve an absolute transparency and high robustness to noise at low authentication rate for $\widehat{AS}_d$ through FEAT, but the approach used in FEAT can also be utilized to achieve any feasible level of the error performance and the authentication rate of $\widehat{AS}_d$ at

cost of transparency.

7.3.2 Security and Robustness

In addition to the strength of the cryptographic primitives used to create the authentication signal, the security of a BTA scheme also depends on the contents of the authentication signal and the embedding scheme (i.e., method for embedding the authentication signal into the message signal). We discuss these security issues in the context of FEAT in the following paragraphs.

Successful Transmission of the Authentication Signal

We consider the transmission to be successfully authenticated if the transmitted signal contains at least one authentication sequence. Due to the low authentication rate in FEAT, Dave may be unable to authenticate Alice's transmission if the transmitted message signal does not contain enough number of frames to embed a *complete* authentication sequence. Therefore, it is imperative to utilize a short digital signature so that the authentication sequence will be short. For this reason, FEAT utilizes a Elliptic Curve Cryptography (ECC) based signature scheme instead of a conventional digital signature scheme (such as RSA-based signatures) [111]. It is well known that ECC-based cryptosystems can provide an equivalent level of security with a much shorter key when compared with conventional cryptosystems. For instance, ECC with a key size of 163 bits provides an equivalent level of security to the signature when compared with RSA with a key size of 1024.

Robustness to Interference

Eve may also attempt to corrupt Alice's authentication signal through selectively jamming the authentication signal. This type of attack, called obstruction of authentication (OOA) jamming, may remain undetected if the transmission power required by Eve to corrupt the

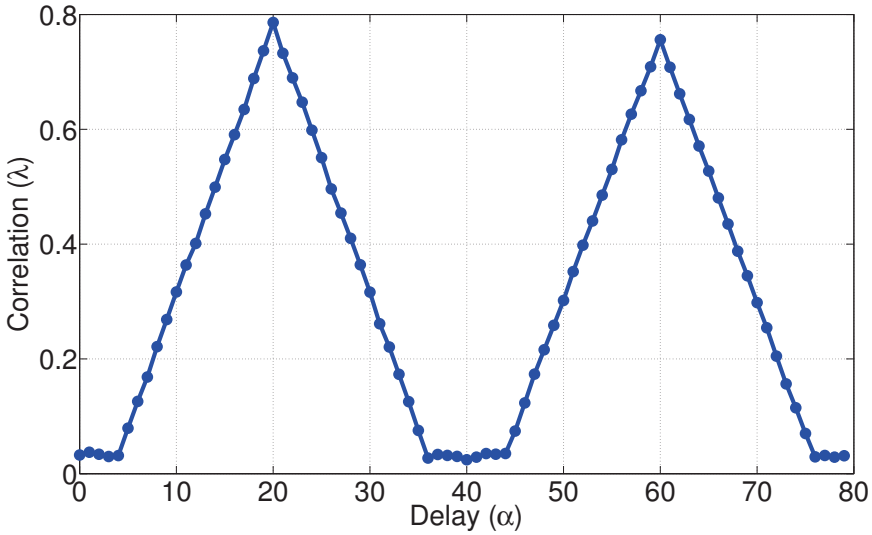


Figure 7.7: Correlation value vs. delay in FEAT.

authentication signal is small as in the case of PHY-layer authentication schemes based on hierarchical modulation [1]. In these schemes, the message signal is embodied by a high-power constellation while the authentication signal is carried on a low-power constellation. An adversary can emit just enough interference to exploit the power difference between the two constellations to disable the decoding of the authentication signal without disabling the decoding of the message signal.

In FEAT, the frequency offset in an OFDM symbol is estimated using the correlation between CP samples and corresponding data samples of the symbol. Hence, only a subset of samples in an OFDM symbol is utilized for estimation of the frequency offset. This means that the change in the correlation among samples in a symbol other than the samples related to CP samples bear no effect on the extraction of the authentication signal in terms of interference to the characteristics used to estimate $\widehat{AS}_d$, i.e., frequency offset. If the received signal contains *mutually exclusive* subsets of CP samples from multiple transmitters, these subsets can be extracted and utilized to estimate frequency offsets in the signals received from multiple transmitters concurrently. In FEAT, the probability that the set of CP samples of the two signals are mutually exclusive can be calculated to be $p_e = 1 - N_c/N_f$. This means that

when we estimate $\lambda = \Lambda(\mathbf{r}, N_f, N_c, \alpha)$ to achieve the symbol synchronization discussed in Section 7.2, we can observe two independent peaks with probability p_e. To illustrate this property, we consider that Dave receives concurrent signals of equal power from Alice with sample offset $\alpha = 20$ and Eve with sample offset $\alpha = 60$. In this case, Figure 7.7 shows the amplitude of λ vs. α for known values of N_f and N_c. We can easily detect the start of the symbol of the signals from Alice at sample 20 and Eve at sample 60. When we synchronize with $\alpha = 20$, we can extract the authentication signal of Alice. On the other hand, when we synchronize with $\alpha = 60$, we extract the authentication signal of Eve. Hence, FEAT is extremely robust against interference from an adversary, and hence OOA jamming attack is not possible for Eve without detection. This means that FEAT enables Dave to detect the identity of Eve easily if Eve utilizes even a small power to jam the message or authentication signals from Alice.

7.4 Performance Evaluation

Based on the performance criteria established in Section 3.5, we evaluate FEAT through comparison with two schemes which represent the existing art of PHY-layer authentication: Authentication Tagging with Modulation (ATM)[1], and Gelato [31].

In FEAT, one bit of the authentication signal is embedded in each frame of the message signal by modifying its frequency offset, i.e., $M = 2$. In ATM, the authentication signal is embedded into the message signal by changing the phase of the QAM message samples. An authentication bit of 1 is embedded by shifting the phase of a QAM sample towards the Q-axis (representing quadrature-phase) by θ. An authentication bit of 0 is embedded by shifting the phase towards the I-axis (representing in-phase) by θ. For the sake of comparison, we embed one authentication bit per frame which means that the phase of all the QAM samples in a frame are shifted in only one direction corresponding to the authentication bit to be embedded. In Gelato, the authentication signal is embedded into the transmitted OFDM

signal by repeating N_a QAM samples over the sub-carriers to generate a cyclo-stationary signature. For the sake of comparison, we embed one authentication bit per frame which means that all the OFDM symbols in a frame carry the same signature. An authentication bit of 1 is embedded by repeating the QAM samples from the first N_a sub-carriers to the next N_a sub-carriers. An authentication bit of 0 is embedded by repeating the QAM samples from the last N_a sub-carriers to the previous N_a sub-carriers.

7.4.1 Overhead

In FEAT, Alice embeds the frequency offset into the message signal through simple vector multiplication over each frame. This means that no significant computation overhead is incurred to include FEAT at Alice. Also, there are no power and message throughput overheads at Alice. Also, no significant overhead is incurred at Bob to use those frequency estimates to estimate the authentication signal. In ATM, no significant computational overhead is needed to embed authentication at Alice along with no power and message throughput overheads. In Gelato, the computation overhead to embed the authentication signal at Alice is non-significant. However, since N_a out of N_u useful sub-carriers are loaded with redundant data samples, the message data-rate is reduced by $\frac{N_a}{N_u} \cdot 100$ %. For instance, with $N_a = 6$ and $N_u = 48$, Alice loses 12.5% of its data-rate.

7.4.2 Compatibility

In the existing standards describing PHY-layer specifications, there is a significant margin allowed for the carrier frequency offset (CFO) in the message signals due to inaccurate oscillators at the transmitters and the receivers. For instance, as per IEEE 802.11g [110], the absolute value of CFO due to an inaccurate oscillator should be less than 25 ppm of the carrier frequency. This means that for transmitted signals at 2.4 GHz, a frequency offset of $\pm$60 kHz is allowed. Also, the preamble structure (inserted in each frame) ensures that a

frequency offset of $2 \cdot 60$ kHz $= 120$ kHz (considering the margin for the oscillator at receiver) can be tolerated by each frame of the message signal. In FEAT, Charlie utilizes the preamble symbols added at the beginning of each frame, and the pilot samples in each symbol of the message signal to estimate and remove the frequency offset. Hence, there is no effect on the error performance of the message signal at Charlie. In ATM, the phase offset can be estimated using the pilot symbols and hence there is no effect on the error performance of the message signal at Charlie. In Gelato, Charlie can demodulate the message signal, but the demodulated signal would not make sense for Charlie since it being the unaware of the presence of the authentication scheme does not know the presence of the repetition of QAM message samples on some of the sub-carriers. Hence, unlike FEAT and ATM, Gelato is not compatible with the unaware receiver.

7.4.3 Message Signal's Error Performance

In FEAT, an intended receiver utilizes the preamble symbols added at the beginning of each frame, and the pilot samples in each symbol of the received signal to estimate and correct the frequency offset. In effect, no change is required in the message decoding procedure at Bob, and the embedding of the authentication signal has no effect on the error performance of the message signal at Bob. In ATM, Bob using its pilot symbols can estimate and remove the phase offset and hence, there is no effect on the error performance of the message signal at Bob. In Gelato, although Bob does not suffer in terms of the error performance of the message signal, the message decoding procedure needs to be modified to discard the data samples at the redundant sub-carriers.

7.4.4 Authentication Signal's Error Performance

We simulate FEAT, ATM and Gelato using Matlab to estimate their error performance at different SNR. With AWGN channel, FEAT performs significantly better than ATM and

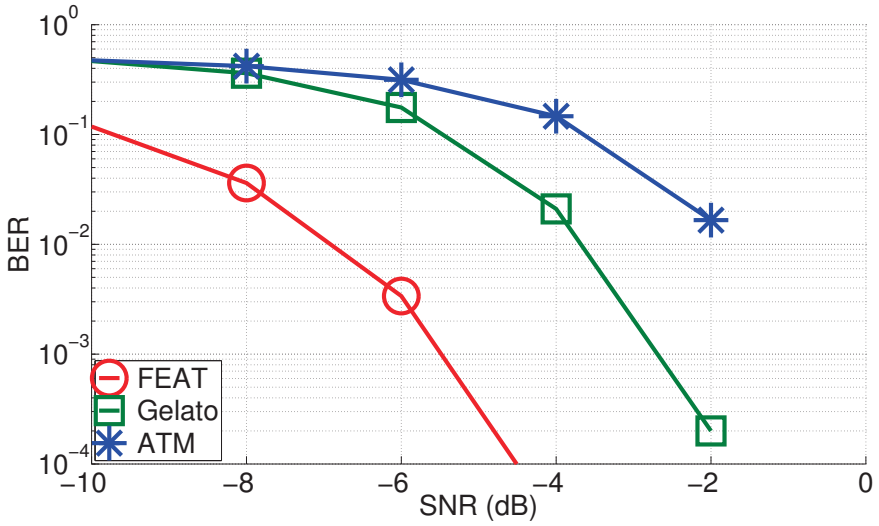


Figure 7.8: Comparison of error performance of $\widehat{AS}_d$ in AWGN channel in FEAT with $f_a = 5$ kHz, Gelato with $N_a = 12$, and ATM with $\theta = \pi/8$, where $N_f = 64, N_c = 16$, and $N_s = 50$

Gelato as shown in Figure 7.8. For instance, at SNR of -6 dB, the BER in FEAT is 0.003 as compared to 0.2 in Gelato, and 0.3 in ATM. We also present the error performance of the authentication signal in a Rayleigh fading channel with 200 Hz doppler shift in Figure 7.9. Recall that since Dave does not have the information of the pilot signals used by Alice, it is not possible for it to counter the channel effects generated due to multipath. Hence, in Figure 7.9, we observe that the BER in ATM is close to 0.5. However, even in these channel conditions, FEAT achieves sufficient BER so that the authentication sequence can be recovered using the error correcting code.

7.4.5 Authentication Rate

By design, in FEAT, ATM as well as Gelato, one bit of authentication signal is embedded into each frame of the message signal. Hence, the authentication rate is equal to the frame rate of the message signal.

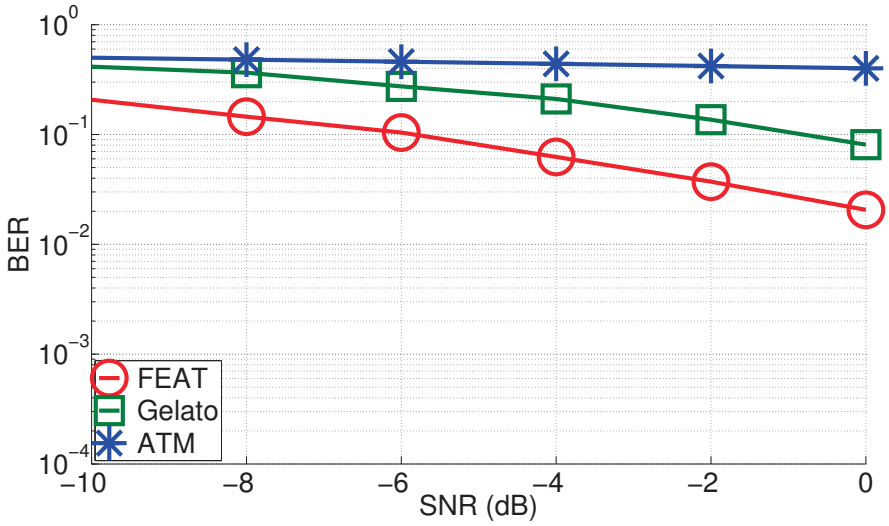


Figure 7.9: Comparison of error performance of $\widehat{AS}_d$ in Rayleigh fading channel in FEAT with $f_a = 5$ kHz, Gelato with $N_a = 12$, and ATM with $\theta = \pi/8$, where $N_f = 64, N_c = 16$, and $N_s = 50$

7.4.6 Authentication of Concurrent Transmissions

FEAT is robust to interference as discussed in Section 7.3. Hence, in presence of concurrent transmissions from Alice and Eve, each of the two can be authenticated at Dave. However, neither Gelato nor ATM can be used to extract the authentication signal from the received signal corrupted by interference from the similar type of signal. In ATM, the phase offsets in the received samples containing the authentication signals from Alice and Eve cannot be separated. In Gelato, in the absence of interference, each OFDM symbol contains one signature. But, when the received signal contains signals from multiple transmitters, multiple cyclostationary signatures can be observed in the received OFDM symbol, and there is no way to extract the authentication signature corresponding to a specific transmitter.

7.4.7 Blind Authentication

At a receiver, after down converting and sampling the received signal, time and frequency synchronization are the first steps to be performed to extract the message signal. A significant amount of work has been done in the field of blind (non-data-aided) parameter estimation, e.g., time and frequency offset estimation, for OFDM signals [112, 113, 114, 107]. Also, it has been shown in [61] that carrier frequency offset (CFO) is an intrinsic characteristic of a transmitter, and it can be used for authentication. Note that the actual CFO of an oscillator in a transmitter usually remains close to a constant value although some variations may be caused due to long life-span, temperature, and other environmental factors. Moreover, it has been shown in [72] that an authentication signal can be extrinsically embedded into the pilot symbols of the message signal in the form of frequency offsets. However, the blind receiver cannot utilize this scheme due to lack of knowledge of the pilot symbols. In FEAT, the authentication signal is embedded into each frame of the message signal using frequency offset such that it can be extracted using the techniques of blind parameter estimation. However, Dave (the blind receiver) needs to know the center frequency and the sampling frequency of the transmitted signal to authenticate the received signal. Gelato with the sample and symbol synchronization mechanism (proposed for FEAT) can be used with the same knowledge as needed in FEAT. In ATM, other than the center frequency and the sampling frequency, the blind receiver also needs to know the modulation being used by the transmitter. In general, the center frequency and the sampling frequency depend on the standard to be utilized to set up the network [105] and hence, their knowledge can be considered to be available a priory. However, modulation schemes depend on the channel conditions between the transmitter and the intended receivers and hence, it is subject to change. This means that FEAT and Gelato enable blind authentication, but ATM does not.

7.4.8 Security

Considering that the contents and the length of the authentication signal in the three schemes are same, we compare the robustness of the scheme in the case where Eve may attempt to corrupt the authentication signal transmitted by Alice, i.e. OOA jamming attack. Since FEAT is the most robust scheme against interference, it is also the most robust scheme against OOA jamming attack. Moreover, since FEAT is the most robust scheme against noise as shown in Figure 7.8, it is also the most secure scheme against incessant jamming.

7.5 Experimental Validation

We conducted a number of experiments using an implementation of FEAT. In the experiments, we used three Universal Software Radio Peripheral (USRP) radios, one each for Alice (transmitter), Bob (aware receiver), and Dave (blind receiver). National Instruments' LabVIEW is utilized as the system-design platform to configure the three USRPs. Alice and Bob use the parameters discussed in IEEE 802.11af standard [105] to communicate with each other. Alice also embeds an authentication signal using FEAT so that Dave is able to authenticate Alice.

7.5.1 Model and Assumptions

The three radios are placed in an indoor environment in such a way that the distance between any two radios is approximately 1 meter. The distances between the radios are limited by the fact that all the radios need to be connected to the computer running the LabVIEW application through network cables. Hence, to obtain a wide range of SNR values (from -10 dB to 10 dB), we add Gaussian noise at Bob and Dave in addition to the channel-induced noise added to the signal transmitted over-the-air. Here, we assume that adding Gaussian noise after receiving the signal is equivalent to increasing the distance between the

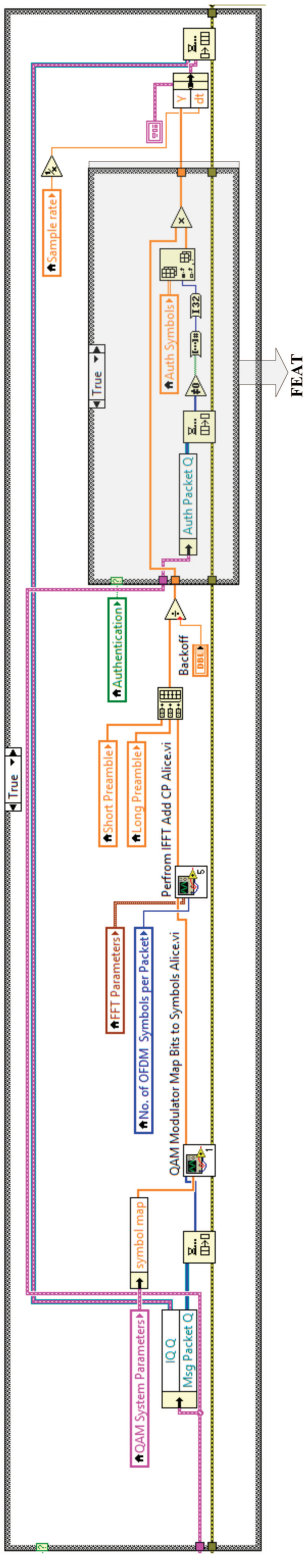


Figure 7.10: LabVIEW VI illustrating the implementation of FEAT.

transmitter and the receivers.

7.5.2 Design

We utilize the following PHY-layer parameters—the center frequency $F_c = 915$ MHz, the sampling frequency $F_s = 1$ MHz, IFFT size $N_f = 64$, the CP size $N_c = 16$, the number of useful sub-carriers $N_u = 52$ (48 for data samples and 4 for pilot samples), and the number of symbols in each frame $N_s = 50$. The preamble consists of four symbols (i.e., $N_p = 4$)—two symbols each for short and long preamble sequence. We utilize quadrature amplitude shift keying (QPSK) as the modulation scheme for the message signal. The data contained in the message signal consists of a time-stamp and a text, and is transmitted without any error correction coding. The authentication signal consists of a set of random bits for synchronization, a time-stamp and a text data without any error correction coding. It is embedded into the message signal using FEAT with $M = 2$ and $f_a = 1$ kHz. Since Bob is the receiver with the knowledge of all the PHY-layer parameters, he demodulates and decodes the received signal. The received message signal is synchronized using a time-stamp, and compared with the transmitted message signal to calculate the BER of the message signal.

Dave extracts the authentication signal by synchronizing with the received signal which is processed in blocks of 1 million samples (i.e., the number of samples received per second). Since the processing overhead needed to achieve synchronization is quite high, the parameters such as IFFT size ($\widehat{N}_f$), CP size ($\widehat{N}_c$), and frame size ($\widehat{N}_s$) are estimated only for the first block of the received samples. During the experiments, we noticed that the value of sample offset ($\widehat{\alpha}$) changes slowly because of the clock mismatch between the hardware platforms. Hence, the sample offset ($\widehat{\alpha}$) and symbol offset ($\widehat{\beta}$) are estimated for each block of received samples. The received authentication signal is synchronized using the synchronization bits, and compared with the transmitted authentication signal to calculate the BER of the authentication signal.

Figure 7.10 shows the LabVIEW VI of Alice illustrating the various steps needed to embed

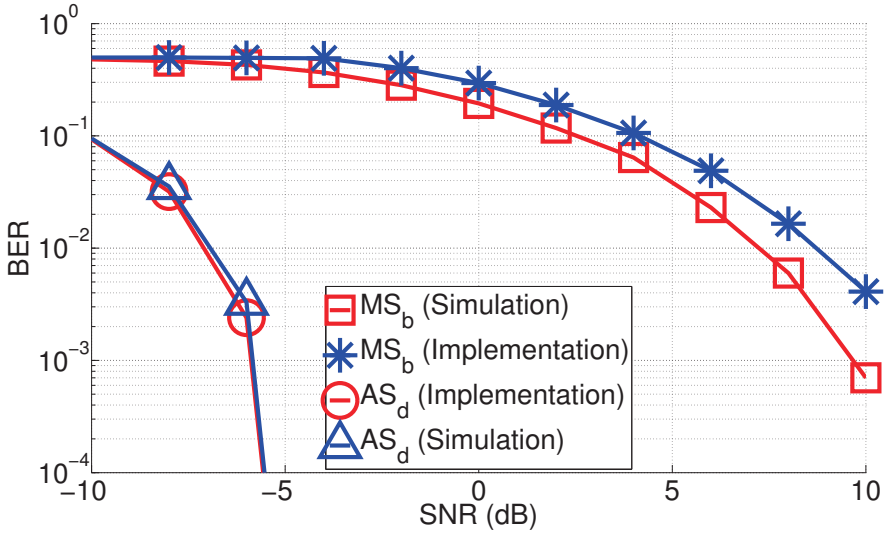


Figure 7.11: Comparison of the error performance of $\widehat{MS}_b$ and $\widehat{AS}_d$ in implementation and simulation of FEAT.

an authentication symbol into a frame of the message signal. The message signal is generated by creating conventional OFDM signals—mapping the message bits to QAM symbols, performing IFFT, adding CP, and adding preamble symbols. To embed the authentication signal, the message signal is multiplied sample-by-sample with a vector which embeds the frequency offset; this process is carried out by the blocks enclosed in the gray box shown in Figure 7.10.

7.5.3 Results

Figure 7.11 shows the error performance of the message signal at Bob ($\widehat{MS}_b$) and the authentication signal at Dave ($\widehat{AS}_d$). The error performance from Matlab simulations with the same PHY-layer parameters are also presented as a benchmark. We observe that the error performance of the USRP implementation is quite close to the error performance obtained from the simulations in the case of the authentication signal. However, the same is not true for the message signal. This result can be explained by recognizing the fact that the channel noise

is Gaussian in the simulations, whereas the channel noise is not truly Gaussian in the over-the-air experiments when the message signal is decoded sample-by-sample. However, when an authentication symbol is estimated by correlating the CP samples of length $N_c \cdot N_s = 800$ with their corresponding data samples of equal length, then the channel noise added in the over-the-air experiments can be considered to be Gaussian for the authentication signal as a result of the central limit theorem.

7.6 Summary

In this chapter, we have defined the BTA problem, and proposed a novel scheme called FEAT that satisfies all of the required criteria of the BTA problem. Through analytical analysis, simulations, and experiments with an USRP-based implementation, we have shown that FEAT is a viable approach for authenticating transmitters even in very harsh channel environments, where the SINR is low and the multipath fading is significant.

Chapter 8

FREE: Frequency Offset Embedding for Crowd-Sourced Blind Authentication of Co-Channel Transmitters

In this chapter, we propose the very first instantiation of crowd-sourced blind authentication of co-channel transmitters (CBAT) called Frequency Offset Embedding for CBAT (FREE) that greatly improves the accuracy and reliability of authentication even when the received signal's quality is very poor and the authentication is performed by blind receivers. This is possible by harnessing the power of crowdsourcing and collaborative processing. To the best of our knowledge, FREE is the only existing PHY-layer authentication scheme that can reliably separate and authenticate waveforms from multiple simultaneous co-channel transmissions. We demonstrate the viability of FREE with experimental results as well as simulation results.

The rest of the chapter is organized as follows. We describe the model and the assumptions

in Section 8.1. We provide the overview of FREE in Section 8.2, and the details of FREE in Section 8.3. We analyze the error performance of FREE in Section 8.4, and evaluate FREE by comparing with the prior art in Section 8.5. We present the implementation results in Section 8.6. Section 8.7 concludes the chapter by highlighting the main contributions.

8.1 Model and Assumptions

We assume that the transmitters utilize CP-based OFDM for their message signals to communicate with their intended receivers. The message signal is transmitted in frames, where each frame contains three parts—a preamble, a header and message data. The preamble in each frame is utilized by the intended receivers to perform time and frequency synchronization. The header contains information regarding the encoding and modulation of the message data. The message data contains the information which needs to be delivered to the intended receiver. The message signaling using OFDM with the preamble is a spectrum-efficient scheme used in almost all modern wireless systems with very high message throughput, and hence the proposed scheme, FREE, to applicable to most modern systems.

Further, we assume that the blind receivers are aware of the fact that OFDM is employed by the transmitters to modulate and transmit the message signals in frames. The blind receivers also know the sampling frequency, the length of the FFT, and the length of CP utilized in the transmitted signals. These parameters are typically standardized as part of an air-interface standard, e.g., IEEE 802.11g. The assumption about the presence of knowledge about the length of the FFT and the length of CP at the blind receivers can be relaxed by complementing FREE with the conventional techniques for OFDM parameter estimation [107]. We also assume that there exists a fine-grained time synchronization among the blind receivers. This can be facilitated by the DFS using the conventional techniques, e.g., distributed primary reference clock and packet-based time synchronization [115, 116].

Further, we assume that the blind receivers receive signals at a very low SINR (e.g., below

0 dB) with significant fading. In such a case, the header in each frame cannot be processed by the blind receivers [106], and hence the blind receivers cannot obtain the PHY-layer parameters utilized to transmit the frames.

8.2 Overview of FREE

We propose a concrete instantiation of the CBAT concept called FREquency offset Embedding for CBAT (FREE). To the best of our knowledge, FREE is the first PHY-layer authentication scheme that addresses all of the following three challenges: (1) authenticating received signals with minimal knowledge of the PHY-layer transmission parameters; (2) authenticating signals emitted simultaneously from *multiple co-channel* transmitters; and (3) reliable authentication of received signals with very low SINR. In FREE, there are three entities—transmitters, blind receivers and a data fusion station (DFS). We provide an overview of the operations at each of these entities in FREE followed by elaborate technical details.

8.2.1 Transmitter

In FREE, a transmitter carries out three major operations. Firstly, the transmitter generates a sequence of frames of the message signal using the conventional OFDM procedures employed in modern communication systems. Secondly, it generates the authentication signal which plays a pivotal role in FREE, because an enforcement entity verifies a transmitter's authentication data contained in the authentication signal to uniquely identify the transmitter. Thirdly, the transmitter embeds the authentication signal into the message signal by modifying the frequency offset of each frame of the message signal. This process is achieved in such a way that the authentication signal does not interfere with the decoding process of the message signal at the intended receivers.

8.2.2 Blind Receiver

At a blind receiver in FREE, there are five major operations. Firstly, the blind receiver down-converts and samples the received signal. Secondly, it computes the decision variable which is utilized to estimate the number of transmitters and their frame boundaries. The decision variable is calculated by first estimating and correcting the frequency offset in the received long training samples of the preamble, and then computing its cross-correlation with a local copy of the long training samples. Thirdly, the blind receiver compares the values in the decision variable to a threshold to determine the number of transmitters and the location of the start of the received frames from the transmitters. This is an important step which enables FREE to address the second challenge of detecting multiple co-channel transmitters. Fourthly, it computes the frequency offset embedded into the frames of the message signal by utilizing the correlation between the CP samples and the corresponding data samples of the OFDM symbols. Note that the frequency offset of a frame is estimated at the blind receivers with only limited knowledge about the transmission parameters. In this way, FREE addresses the first challenge of blind authentication of the transmitters. Finally, the blind receiver communicates the estimated values of the frequency offsets to the DFS.

8.2.3 DFS

The DFS aggregates the values of the frequency offsets reported from multiple blind receivers. It utilizes the aggregated frequency offsets to estimate the authentication signal for each transmitter. The collaboration of the blind receivers enabled by the DFS significantly improves the performance of the estimated authentication signal, and addresses the third challenge of robust authentication at very low SINR.

8.3 Details of FREE

In the following discussions, we assume that the transmitters and blind receivers are uniformly distributed in a hexagonal cell. At a particular communication channel, represented by F_c, the number of transmitters is represented as N_t. The i^{th} transmitter is represented as $\mathsf{Tx_i}$, where $i = 1, 2, \cdots, N_t$. Also, there are N_b blind receivers receiving the signals in the communication channel. The j^{th} blind receiver is represented as $\mathsf{BRx_j}$, where $j = 1, 2, \cdots, N_b$. The operations at each of the three entities, i.e., $\mathsf{Tx_i}$, $\mathsf{BRx_j}$, and DFS, are discussed below.

8.3.1 Operations at $\mathsf{Tx_i}$

Generation of a Frame of the Message Signal

The message signal to be transmitted by the $\mathsf{Tx_i}$ in k^{th} frame is represented by $\mathbf{b_{ik}}$. The $\mathbf{b_{ik}}$ is a sequence of message data bits which are assumed to be statistically independent and identically distributed with zero mean. The $\mathbf{b_{ik}}$ is encoded using a convolution code of rate R_{ik}, and modulated to message data symbols using the quadrature amplitude modulation (QAM) of order M_{ik}.

To generate one OFDM symbol, the available frequency band is divided into N_f sub-carriers. Out of the N_f symbols corresponding to the N_f sub-carriers, N_u symbols are assigned to message data, and N_p symbols are assigned to message pilot symbols. The rest of the $N_f - N_u - N_p$ symbols are set to zero. The N_f samples are generated by taking the Inverse Fast Fourier Transform (IFFT) of the N_f symbols. The last N_c samples out of the N_f samples are repeated at the beginning of the N_f samples as the CP to generate an OFDM symbol. The total number of OFDM symbols in a frame is represented by N_a. The total number of samples in a frame is given by $N_o = N_a \cdot (N_f + N_c)$.

Further, the message header of the frame, which contains the values of R_{ik} and M_{ik}, is generated. The message header is appended at the start of the OFDM symbols. Also, the

message preamble is appended at the start of the message header. We employ the message preamble structure utilized in multiple standards, e.g., IEEE 802.11g and IEEE 802.11af [110]. It consists of 10 repetitions of a set of short training samples, a CP guard interval, and two repetitions of a set of long training samples. The number of samples in the set of short training samples is equal to $N_s = N_f/4$; in the CP guard interval is equal to $N_f/2$; and in the set of long training samples is equal to N_f. Finally, the sequence of the samples in k^{th} frame is represented by $\mathbf{s_{ik}}$ of length N_o.

Generation of the Authentication Signal

The authentication signal contains a time-stamp, represented by T_{si}, a regulator-assigned identity of the transmitter, represented by I_i, the frequency channel allowed for transmission, represented by F_c, registered location of the transmitter, represented by L_i, and hours of operation authorized by the regulator, represented by T_{hi}. The sequence of bits, represented by $\mathbf{a_{di}} = \{T_{si}, I_i, F_c, L_i, T_{hi}\}$, is digitally signed. The signature of $\mathbf{a_{di}}$ is represented by $sign(\mathbf{a_{di}})$. This signature can be generated using a conventional digital signature scheme. We will not elaborate any further on the signature scheme, as it is outside the scope of this paper. The sequence of bits given by $\{\mathbf{a_{di}}, sign(\mathbf{a_{di}})\}$ is channel-coded using convolution coding. Finally, a sequence of authentication synchronization bits are appended to generate the authentication signal, represented by $\mathbf{a_i}$, of K_a bits.

Embedding of the Authentication Signal into the Frames of the Message Signal

In FREE, the authentication signal $\mathbf{a_i}$ is mapped to authentication symbols, represented by $\mathbf{v_i}$, using non-return-to-zero (NRZ) encoding. This means that $\mathbf{v_i}[k] = +1$ if $\mathbf{a_i}[k] = 1$, and $\mathbf{v_i}[k] = -1$ if $\mathbf{a_i}[k] = 0$, for $k = 0, 1, \cdots, K_a$. Note that we utilize the notation $\mathbf{a_i}[k]$ to represent the k^{th} element of the sequence $\mathbf{a_i}$. In FREE, one authentication symbol is embedded into one frame of the message signal by inducing a frequency offset into the samples of the frame. In the k^{th} frame, the embedded frequency offset (EFO) is denoted by f_{aik},

and is computed as $f_{aik} = \mathbf{v_i}[k] \cdot f_a$, where f_a is the constant parameter set by the DFS and utilized by all the transmitters. The parameter, f_a, plays an important role in determining the robustness of the embedded authentication signal against noise (see Section 8.4.1).

Hence, for $n = 0, 1, \cdots, N_o - 1$, each sample of the k^{th} frame of the embedded signal (i.e., the message signal embedded with the authentication signal) in the baseband is given by $\mathbf{x_{ik}}[n] = \mathbf{s_{ik}}[n] \cdot e^{j2\pi \frac{f_{aik}}{F_s} n}$, where F_s is the sampling frequency. Finally, the embedded signal is up-converted to the carrier frequency F_c and transmitted. Assuming that frequency offset induced due to the inaccurate oscillator at $\mathsf{Tx_i}$ is f_{ti}, each sample of the k^{th} frame of the transmitted signal is given by $\mathbf{y_{ik}}[n] = \mathbf{x_{ik}}[n] \cdot e^{j2\pi \frac{f_{ti}}{F_s} n}$, for $n = 0, 1, \cdots, N_o - 1$. We assume that the transmitted signal at $\mathsf{Tx_i}$, represented by $\mathbf{y_{ti}}$, contains more frames than K_a so that at least one authentication sequence of K_a bits is successfully transmitted.

8.3.2 Operations at $\mathsf{BRx_j}$

Down-Conversion and Sampling

In FREE, the $\mathsf{BRx_j}$ down-converts and samples the received signal in the considered frequency band F_c. The n^{th} sample of the received signal with N_r discrete samples is represented by $\mathbf{r_{rj}}[n] = \sum_{i=1}^{N_t} h_{ji} \cdot \mathbf{y_{ti}}[n + \alpha_{ji}] \cdot e^{j2\pi \frac{f_{rj}}{F_s} n} + z_{nj}$, where h_{ji} represents the free space path loss between the $\mathsf{Tx_i}$ and the $\mathsf{BRx_j}$, α_{ji} represents the start of the first received frame from the $\mathsf{Tx_i}$, f_{rj} represents the frequency offset induced due to the inaccurate oscillator at $\mathsf{BRx_j}$, and z_{nj} represents the white Gaussian noise with mean equal to zero and variance equal to σ_z^2. In this paper, we do not consider fading to simplify the discussions below.

Computation of Decision Variable

Assuming the start of a frame in the received signal to be $\widetilde{\alpha} = 0, 1, \cdots, N_o - 1$, the $\mathsf{BRx_j}$ performs the following.

1. The $\mathsf{BRx_j}$ segments the received samples into $\widetilde{K}_r$ frames of length N_o samples, where $\widetilde{K}_r = \lfloor (N_r - \widetilde{\alpha})/N_o \rfloor$. Here, $\lfloor v \rfloor$ denotes the largest integer less than or equal to v. For all $k = 0, 1, \cdots, \widetilde{K}_r$, the k^{th} frame in the segmented samples is represented as $\mathbf{r_{sj\widetilde{\alpha}k}}[n] = \mathbf{r_{rj}}[k \cdot N_o + \widetilde{\alpha} + n]$, $\forall n = 0, 1, \cdots, N_o - 1$. For $k = 0, 1, \cdots, \widetilde{K}_r$, the $\mathsf{BRx_j}$ performs the following operations.

   (a) The $\mathsf{BRx_j}$ computes the auto-correlation induced due to the short training samples of the message preamble as $P_{sj\widetilde{\alpha}k} = \frac{1}{9 \cdot N_s} \sum_{n=0}^{9 \cdot N_s - 1} \mathbf{r^*_{sj\widetilde{\alpha}k}}[n] \cdot \mathbf{r_{sj\widetilde{\alpha}k}}[N_s + n]$, where v^* denotes the complex conjugate of v. The estimated coarse frequency offset is obtained as $\widetilde{f}_{sj\widetilde{\alpha}k} = \frac{F_s}{2\pi N_s} \angle P_{sj\widetilde{\alpha}k}$, where $\angle v$ denotes the polar angle of the complex number v. The samples corresponding to the long training samples of the message preamble are extracted from the frame, and are adjusted for the coarse estimate of the frequency offset as $\mathbf{r_{tj\widetilde{\alpha}k}}[n] = \mathbf{r_{sj\widetilde{\alpha}k}}[3 \cdot N_f + n] \cdot e^{-j2\pi \widetilde{f}_{sj\widetilde{\alpha}k}(3 \cdot N_f + n)/F_s}$, $\forall n = 0, 1, \cdots, 2 \cdot N_f - 1$.

   (b) The $\mathsf{BRx_j}$ computes the auto-correlation between the long training samples of the message preamble as $P_{tj\widetilde{\alpha}k} = \frac{1}{N_f} \sum_{n=0}^{N_f - 1} \mathbf{r^*_{tj\widetilde{\alpha}k}}[n] \cdot \mathbf{r_{tj\widetilde{\alpha}k}}[N_f + n]$. The estimate of fine frequency offset is obtained as $\widetilde{f}_{tj\widetilde{\alpha}k} = \frac{F_s}{2\pi N_f} \angle P_{tj\widetilde{\alpha}k}$. The long training samples of the message preamble are adjusted for the fine estimate of the frequency offset as $\mathbf{r_{lj\widetilde{\alpha}k}}[n] = \mathbf{r_{tj\widetilde{\alpha}k}}[n] \cdot e^{-j2\pi \widetilde{f}_{tj\widetilde{\alpha}k}(3 \cdot N_f + n)/F_s}$, $\forall n = 0, 1, \cdots, 2 \cdot N_f - 1$.

   (c) The cross-correlation between the local copy of the long training symbols of the message preamble, represented by $\mathbf{p_l}$, and the received samples corresponding to the message preamble is computed as $\mathbf{\Phi_{j\widetilde{\alpha}}}[k] = \frac{1}{2 \cdot N_f} \sum_{n=0}^{2 \cdot N_f - 1} \mathbf{p^*_l}[n] \cdot \mathbf{r_{lj\widetilde{\alpha}k}}[n]$.

2. The decision variable is computed by averaging over the $\widetilde{K}_r$ frames as $\mathbf{\Psi_j}[\widetilde{\alpha}] = \sum_{k=0}^{\widetilde{K}_r - 1} \left| \mathbf{\Phi_{j\widetilde{\alpha}}}[k] \right| / \widetilde{K}_r$, where $|v|$ denotes the absolute value of v.

Transmitter Detection and Time Synchronization

The $\mathsf{BRx_j}$ performs the following heuristic algorithm to detect the number of transmitters, represented by $\widehat{N}_{tj}$; and the start of the first received frame from the transmitters, represented by $\widehat{\alpha}_{ji}$, for $i = 1, 2, \cdots, \widehat{N}_{tj}$.

1. Set $i = 1$, and $\mathbf{\Lambda_{ji}} = \mathbf{\Psi_j}$.
2. Compute the mean of the decision variable, $\lambda_{ji} = \sum_{\widetilde{\alpha}=0}^{N_o-1} \mathbf{\Lambda_{ji}}[\widetilde{\alpha}]/N_o$, and the threshold, $\tau_\lambda = 2 \cdot \lambda_{ji}$.
3. Compute $\widehat{\alpha}_{ji} = \text{argmax}_{\widetilde{\alpha}} \mathbf{\Lambda_{ji}}[\widetilde{\alpha}]$.
4. If $\mathbf{\Lambda_{ji}}[\widehat{\alpha}_{ji}] > \tau_\lambda$, set $\widehat{\alpha}_{ji}$ as the start of the frame from i^{th} transmitter; otherwise, set $\widehat{N}_{tj} = i - 1$ and exit.
5. Set $\mathbf{\Lambda_{j(i+1)}} = \mathbf{\Lambda_{ji}}$.
6. Set $\mathbf{\Lambda_{j(i+1)}}[\widetilde{\alpha}] = \lambda_{ji}$ for $\widetilde{\alpha} = \widehat{\alpha}_{ji}, (\widehat{\alpha}_{ji} - N_f) \bmod N_o$, and $(\widehat{\alpha}_{ji} + N_f) \bmod N_o$.
7. Set $i = i + 1$ and go back to Step 2.

The number of frames received at $\mathsf{BRx_j}$ is obtained as $\widehat{K}_{rj} = \lfloor (N_r - \widehat{\alpha}_{j1})/N_o \rfloor$. The samples corresponding to the frame, $k = 0, 1, \cdots, \widehat{K}_{rj} - 1$, of the detected transmitter, $i = 1, \cdots, \widehat{N}_{tj}$, are represented by $\mathbf{r_{fjik}}[n] = \mathbf{r_{rj}}[k \cdot N_o + \widehat{\alpha}_{ji} + n]$ for $n = 0, 1, \cdots, N_0 - 1$.

Parameter Estimation

For each of the detected transmitters, $i = 1, 2, \cdots, \widehat{N}_{tj}$, $\mathsf{BRx_j}$ estimates the following three parameters for each of the frames, $k = 0, 1, \cdots, \widehat{K}_{rj}$.

1. *Time of Arrival*: Assume that the $\mathsf{BRx_j}$ starts the detection at time represented by T_{rj}. Hence, the time of arrival of the k^{th} frame of the i^{th} transmitter, represented by

$\widehat{t}_{jik}$, is computed as $\widehat{t}_{jik} = T_{rj} + T_s \cdot \widehat{\alpha}_{ji} + T_s \cdot N_o \cdot k$, where T_s is the sampling time. The sampling time is computed as $T_s = 1/F_s$.

2. *Embedded Frequency Offset*: Note that the true frequency offset of the received signal at the $\mathsf{BRx_j}$ from the $\mathsf{Tx_i}$ is given by $f_{ojik} = f_{aik} + f_{ti} + f_{rj}$. Hence, the frequency offsets in the frames of the message signal corresponding to one authentication sequence has one constant part, represented by $f_{mji} = f_{ti} + f_{rj}$, and a variable part, given by the EFO f_{aik}. Through the following steps, $\mathsf{BRx_j}$ obtains the estimate of f_{mji}, represented by $\widehat{f}_{mji}$, and the estimate of f_{aik}, represented by $\widehat{f}_{ajik}$. The estimate of the frequency offset in the k^{th} frame from i^{th} transmitter is obtained using the correlation given by

$$P_{fjik} = \frac{1}{N_a \cdot N_c} \sum_{l=0}^{N_a-1} \sum_{n=0}^{N_c-1} \mathbf{r^*_{fjik}}[l \cdot (N_f + N_c) + n] \cdot \mathbf{r_{fjik}}[l \cdot (N_f + N_c) + N_f + n].$$

The frequency offset for k^{th} frame is estimated as $\widehat{f}_{ojik} = \frac{F_s}{2\pi N_f} \angle P_{fjik}$. The constant part of the frequency offset is computed as $\widehat{f}_{mji} = \sum_{k=0}^{\widehat{K}_{rj}-1} \widehat{f}_{ojik} / \widehat{K}_{rj}$. The estimate of the EFO is obtained as $\widehat{f}_{ajik} = \widehat{f}_{ojik} - \widehat{f}_{mji}$.

3. *Authentication Signal to Interference and Noise Ratio (ASINR)*: The $\mathsf{BRx_j}$ utilizes the auto-correlation in the received samples to estimate the message signal to interference and noise ratio (MSINR) of the received frame [117]. For the k^{th} frame from i^{th} transmitter, the $\mathsf{BRx_j}$ computes $E_{fjik} = \frac{1}{N_a \cdot N_c} \sum_{l=0}^{N_a-1} \sum_{n=0}^{N_c-1} \left| \mathbf{r^*_{fjik}}[l \cdot (N_f + N_c) + n] \right|^2$. The $\mathsf{BRx_j}$ calculates an estimate of the MSINR corresponding to k^{th} frame as $\widehat{\rho}_{jik} = \left|P_{fjik}\right| / (E_{fjik} - \left|P_{fjik}\right|)$. The estimate of the ASINR corresponding to k^{th} frame is computed as $\widehat{\sigma}_{jik} = \frac{4\pi^2 \cdot N_c N_a N_f^2 \cdot f_a^2}{F_s^2} \cdot \frac{\widehat{\rho}_{jik}^2}{2 \cdot \widehat{\rho}_{jik} + 1}$.

Parameter Communication

Having computed the estimated values of the parameters, the $\mathsf{BRx_j}$ communicates the set of the estimated values, represented by $\mathbf{D_{jik}} = \{\widehat{t}_{jik}, \widehat{f}_{ajik}, \widehat{\sigma}_{jik}\}$ for $i = 1, 2, \cdots, \widehat{N}_{tj}$; and $k = 0, 1, \cdots, \widehat{K}_{rj}$ to the DFS.

8.3.3 Operations at the DFS

For a particular center frequency F_c, the DFS receives the set of estimated values $\mathbf{D_{jik}}$ from the blind receivers, $\forall j = 1, 2, \cdots, N_b$; $i = 1, 2, \cdots, \widehat{N}_{tj}$; and $k = 0, 1, \cdots, \widehat{K}_{rj} - 1$. The DFS determines the number of transmitters to be $\widehat{N}_t = \max_{j=1,2,\cdots,N_b}(\widehat{N}_{tj})$. For each detected transmitter, the DFS performs the following operations.

Data Aggregation

Using the time of arrival $\widehat{t}_{jik}$, the DFS finds all the $\mathbf{D_{jik}}$ from the blind receivers corresponding to the same k^{th} transmitted frame. The set of frequency offsets corresponding to the k^{th} frame is represented by $\{\widehat{f}_{ajik}\}$, for $j = 1, 2, \cdots, N_b$; and the corresponding set of ASINR is represented by $\{\widehat{\sigma}_{jik}\}$, for $j = 1, 2, \cdots, N_b$. The DFS merges these frequency offsets to obtain $\widehat{f}_{aik} = \sum_{j=1}^{N_b} w_{lj}\widehat{\sigma}_{jik}\widehat{f}_{ajik} / \sum_{j=1}^{N_b} w_{lj}\widehat{\sigma}_{jik}$, where w_{lj} represents a weight value corresponding to the trustworthiness of the reported values of the blind receiver, $\mathsf{BRx_j}$, as perceived by the DFS. Without these "trustworthiness" weights, the data aggregation becomes the conventional maximal ratio combining (MRC) [118]. The trustworthiness weights are employed to make the data aggregation algorithm of FREE robust against Byzantine attacks [74]. In the context of FREE, a Byzantine attack represents a scenario in which a subset of the blind receivers (called rogue blind receivers) provides intentionally incorrect estimates of the EFOs to the DFS. In FREE, the DFS utilizes the trustworthiness weights to differentiate the "trustworthiness" of the blind receivers. After the *successful* completion of each CBAT process by the DFS, these trustworthiness weights are adjusted based on the accuracy of the EFO reported by each blind receiver (see Section 8.3.3).

Decoding

The DFS maps $\widehat{f}_{aik}$ to the authentication symbol, represented by $\widehat{\mathbf{v}}_\mathbf{i}[k]$, such that $\widehat{\mathbf{v}}_\mathbf{i}[k] = -1$, if $\widehat{f}_{aik} < 0$; and $\widehat{\mathbf{v}}_\mathbf{i}[k] = +1$, if $\widehat{f}_{aik} \geq 0$. The estimate of the authentication signal, $\widehat{\mathbf{a}}_\mathbf{i}$,

is obtained by demodulating the authentication symbols to corresponding bits using NRZ decoding. This implies that an estimated authentication bit, $\widehat{\mathbf{a}}_\mathbf{i}[k] = 0$, if $\widehat{\mathbf{v}}_\mathbf{i}[k] = -1$; and $\widehat{\mathbf{a}}_\mathbf{i}[k] = 1$, if $\widehat{\mathbf{v}}_\mathbf{i}[k] = +1$.

The cross-correlation between the local copy of the authentication synchronization bits and the received sequence of authentication bits, $\widehat{\mathbf{a}}_\mathbf{i}$, is utilized to estimate the start of the authentication data bits. Further, the estimate of the correct authentication data sequence is obtained by detecting and correcting any bit errors in the sequence of the authentication data bits using convolution coding.

Verification

The estimates of the contents of the authentication sequence are extracted, and then verified using a digital signature verification procedure.

Weight Updates

The blind receivers' trustworthiness weights are initialized as $w_{lj} = 1$, for $l = 0$ and $j = 1, 2, \cdots, N_b$. Note that due to its unique properties, a digital signature can only be correctly verified if all the bits in the digital signature are estimated correctly. This means that if the digitally signed authentication sequence is verified as valid, then the DFS knows exactly all the authentication bits; and the corresponding authentication symbols, and the values of the true EFOs, f_{aik}. Hence, after the successful verification of $(l+1)^{th}$ authentication sequence, the true values of f_{aik} can be utilized as feedback information for updating the trustworthiness weights using the following procedure. For $j = 1, 2, \cdots, N_b$,

1. Compute $w_{aj} = (1/2 - \widehat{\sigma}_{jik} \sum_{k=0}^{K_a-1} (\widehat{f}_{ajik} - f_{aik})/K_a)^2$. Note that the value of w_{aj} is close to 0 if the parameters, $\widehat{f}_{ajik}$ and $\widehat{\sigma}_{jik}$, are correctly reported.

2. Compute $w_{bj} = \frac{1}{w_{aj}} / (\sum_{j=1}^{N_b} \frac{1}{w_{aj}})$.

Table 8.1: PHY-layer parameters of the message signal used in the analysis of error performance of FREE.

Length of FFT used in each OFDM symbol, N_f	64
Number of sub-carriers with data symbols, N_u	48
Number of sub-carriers with pilot symbols, N_p	4
Length of CP in each OFDM symbol, N_c	16
Number of OFDM symbols in each frame, N_a	20
Order of QAM modulation, M_{ik}, $\forall i, k$	4
Rate of convolution coding, R_{ik}, $\forall i, k$	1/2
Sampling frequency (in MHz), F_s	5

3. Update the weight as $w_{(l+1)j} = (w_{bj} + lw_{lj})/(l+1)$.

8.4 Analysis

When evaluating FREE through simulations, we used the parameters given in Table 8.1. These parameter values were obtained from [110], and they are used in OFDM-based 802.11 systems. The authentication signal is embedded into the message signal using the EFO value of $f_a = 2.5$ kHz.

8.4.1 Error Performance

One Transmitter

When there is only one transmitter in a particular channel, there is no co-channel interference. In this scenario, the mean square error (MSE) of the estimate of the EFO $\widehat{f}_{aik}$ at the DFS in FREE can be lower bounded [109]. Using the lower bound on the MSE,

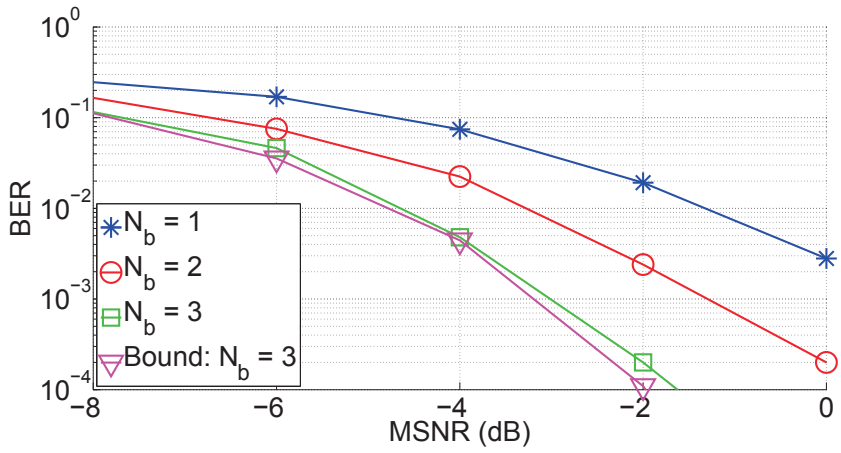


Figure 8.1: Effect of crowd-sourcing on the BER performance of the authentication signal in FREE.

the lower bound on the bit error rate (BER) of the authentication signal is computed as $P_{ea} = \frac{1}{2} \cdot \text{erfc}\left(\sqrt{4\pi^2 N_a N_c N_b \cdot \frac{N_f^2 f_a^2}{F_s^2} \cdot \frac{\rho_{ik}^2}{2\rho_{ik}+1}}\right)$, where ρ_{ik} represents the average of the message signal to noise ratio (MSNR) received at the blind receivers, and erfc represents the complementary error function.

Figure 8.1 presents the BER of the authentication signal vs. MSNR curves in FREE for different numbers of blind receivers, N_b. In the figure, we observe that as N_b increases, the BER of the authentication signal decreases. At the BER of 10^{-2}, the DFS can achieve approximately a 3 dB gain in MSNR by increasing the number of blind receivers from 1 to 3. In the figure, we also observe that the magenta curve with triangle markers representing the theoretical BER lower bound given by P_{ea} closely matches the curve representing the simulated BER when $N_b = 3$.

Multiple Co-channel Transmitters

In FREE, the EFO in an OFDM symbol is estimated using the correlation properties between the CP samples and corresponding data samples. This means that the change in the

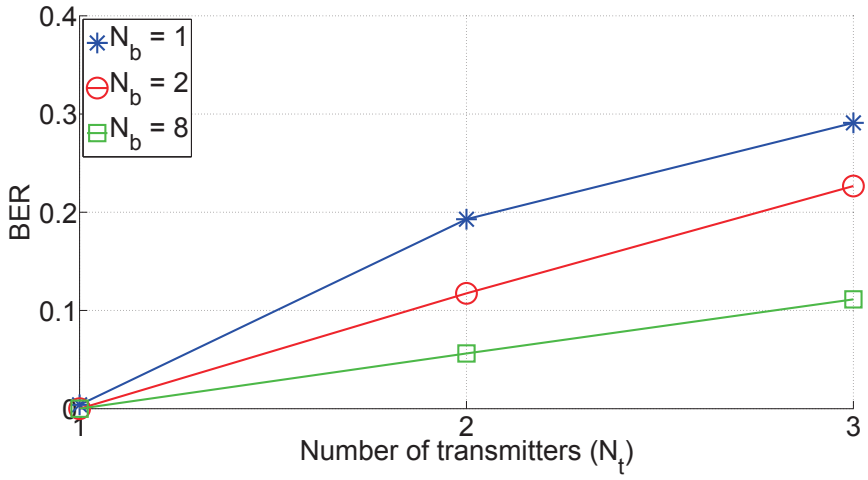


Figure 8.2: Effect of co-channel transmitters and crowd-sourcing on the BER performance of the authentication signal in FREE.

correlation among those samples due co-channel transmissions may hamper the estimation of the EFO in two ways. Firstly, if the received signal contains *mutually exclusive* subsets of CP samples from multiple transmitters (i.e., when the CP samples from multiple transmitters do not overlap), the interference from samples of one transmitter can be considered Gaussian noise to the CP samples of the other transmitters. Note that an OFDM signal is Gaussian in nature. In this case, these subsets of CP samples and corresponding data samples can be robustly extracted, and utilized to estimate the EFOs in the signals received from multiple transmitters concurrently. Secondly, when the CP samples from multiple transmitters overlap, the correlation between the samples (induced due to the EFO) of one transmitter affects the correlation between the received CP samples and corresponding data samples of other transmitters. In this case, the estimation of the EFOs in the signals received from multiple transmitters is significantly degraded.

However, in FREE, as a result of collaborative detection by multiple blind receivers, the transmitted signals are received at different channel gains at different blind receivers. Hence, signals simultaneously emitted from multiple co-channel transmitters can be detected and their EFOs can be extracted at different blind receivers. By aggregating the data from these

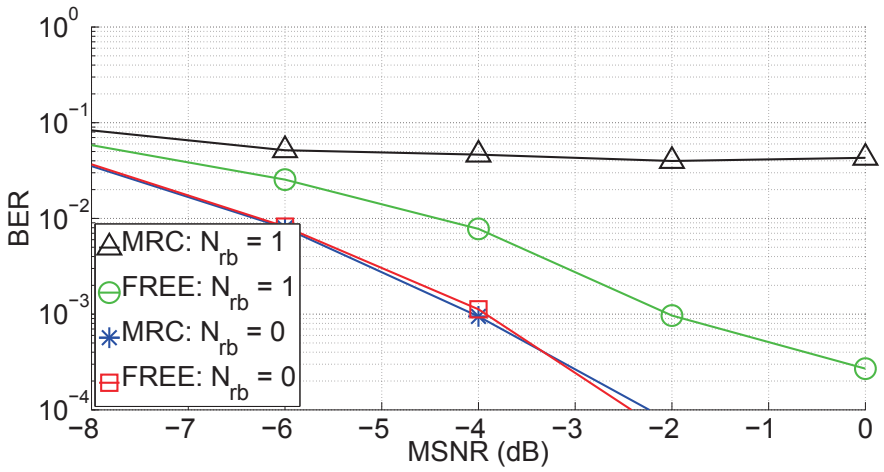


Figure 8.3: Effect of byzantine attack on FREE and MRC.

blind receivers, it is possible for the DFS to *authenticate the multiple co-channel transmitters*. This is the most important characteristic of FREE that distinguishes it from prior art. Note that, in this case, it is prohibitively complex to find a theoretical estimate of the authentication signal's BER.

Figures 8.2 presents the average BER of the authentication signal vs. the number of co-channel transmitters, N_t, for different values of the number of blind receivers, N_b, when the MSNR is 0 dB. In the figure, we observe that if there is only one transmitter in a channel, then the authentication signal's BER is very small ($< 10^{-2}$). However, when there are multiple transmitters in the channel (i.e., $N_t > 1$), the interference from the other transmitters significantly degrade the authentication signal's BER. We can also observe that the impact of interference (from the other co-channel transmitters) on the BER can be mitigated through crowd-sourced BTA, i.e., by increasing N_b.

8.4.2 Security: Robustness Against the Byzantine Attack

Figure 8.3 shows the authentication signal's BER vs. MSNR curves for FREE in two scenarios—when the number of *rogue* blind receivers, N_{rb}, is one or zero. The total number

of blind receivers is four, i.e., $N_b = 4$. The figure also presents the same curves for the conventional MRC algorithm as a benchmark. In both schemes, a rogue blind receiver does not report the correct estimates of EFO for half of the total number of EFO estimates; the reported values of the EFO are randomly selected from the range of possible values of EFO at a rogue blind receiver. In the figure, we observe that when $N_{rb} = 0$, the BER curves of MRC and FREE are almost identical. However, when $N_{rb} = 1$, FREE clearly outperforms MRC. This result can be attributed to the fact that MRC has no mechanism for mitigating the impact of inaccurately reported values. Note that the feedback information utilized by FREE's algorithm is the primary contributor to FREE's robustness against Byzantine attacks.

8.5 Performance Evaluation

To evaluate FREE, we compared its performance with two benchmarks—viz., *FEAT* and *Gelato* [31]. In all the three scheme, FREE, FEAT and Gelato, the message signal is generated using the parameters shown in Table 8.1. In FREE and FEAT, one bit of the authentication signal is embedded in each frame of the message signal by modifying its frequency offset. The authentication bit 1 and 0 are embedded by inducing the frequency offsets +2.5 kHz and −2.5 kHz, respectively. Note that the procedures for embedding the authentication signal are the same in FREE and FEAT, but the procedure for extracting the authentication signal is *different*. In Gelato, the authentication signal is embedded into the transmitted OFDM signal by repeating 12 message data symbols over the sub-carriers to generate a cyclo-stationary signature.

To the best of our knowledge, FEAT and Gelato are the only existing schemes that can be utilized for blind authentication which is the first attribute of an ideal CBAT scheme discussed in Section 1.2.3. Hence, we compare the three schemes based on the remaining four attributes as follow.

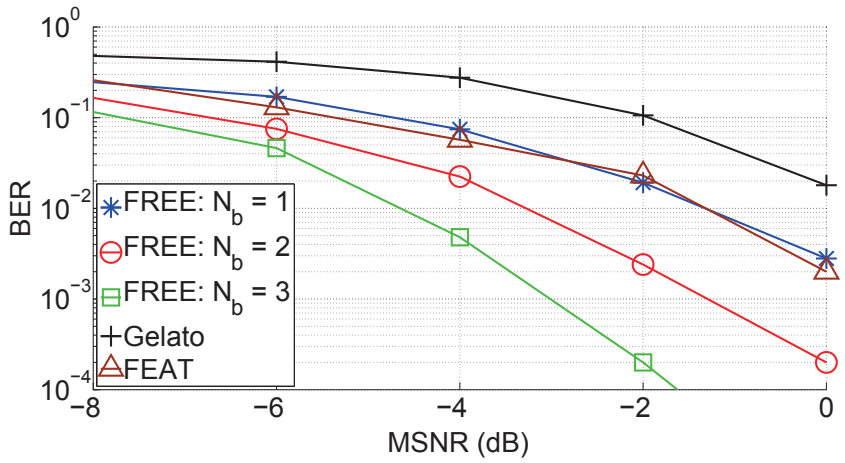


Figure 8.4: Comparison between FREE, FEAT and Gelato.

8.5.1 Overhead

In FREE and FEAT, the transmitter embeds the frequency offset into the message signal through simple vector multiplication over each frame. This means that no significant computational overhead is incurred at the transmitter. Also, there are no changes in transmission power and message throughput at the transmitter. In Gelato, the computational overhead to embed the authentication signal at the transmitter is non-significant. However, since 12 out of 48 data sub-carriers are loaded with redundant data symbols, the message throughput is reduced by 25%.

8.5.2 Compatibility

In the existing air-interface standards, there is a margin of error allowed for the carrier frequency offset (CFO) in the message signals due to inaccurate oscillators at the transmitters and the receivers. For instance, as per the IEEE 802.11g standard [110], a CFO of less than 25 ppm of the carrier frequency is allowed. This means that for signals transmitted at 2.4 GHz, a CFO of ± 60 kHz is allowed. FREE and FEAT are designed to function correctly within the allowed CFO margin of error, and hence they are *compatible* with intended receivers. On

the other hand, Gelato is not compatible. For correct decoding of the message data, Gelato requires the message decoding procedure at an intended receiver to be modified to discard the redundant data samples at the 12 sub-carriers.

8.5.3 Message Signal's Error Performance

In today's modern communication systems, an intended receiver utilizes the message preamble, and the pilot samples in each frame of the received signal to estimate and correct the frequency offset. In effect, FREE and FEAT require no change in the message decoding procedure at an intended receiver, and moreover, they do not impact the BER performance of the message signal at the intended receiver. Also, Gelato does not negatively impact an intended receiver's BER performance.

8.5.4 Authentication Signal's Error Performance

In Figure 8.4, we compare the BER performance of the authentication signal in FREE, FEAT and Gelato. We can observe that the collaborative authentication performed by multiple blind receivers provides FREE with a noticeable advantage over the other two schemes in terms of the BER. This advantage becomes more pronounced as the number of blind receivers, N_b, is increased.

8.5.5 Authentication Rate

In all three schemes, by design, one bit of the authentication signal is embedded into each frame of the message signal. Hence, the authentication rate is equal to the frame rate of the message signal.

8.5.6 Authentication of Concurrent Transmissions

In real-world enforcement scenarios, a rogue transmitter's signal may be weaker than that of compliant transmitters (when measured by an enforcement entity). Moreover, the enforcement entity may need to authenticate signals being transmitted simultaneously from multiple (possibly rogue) co-channel transmitters. Note that the concurrent transmissions hamper the decoding of their authentication signals in two ways—sample-by-sample interference and authentication signature interference. Since an OFDM signal is Gaussian in nature, sample-by-sample interference from samples of one transmitter can be considered Gaussian noise to the other transmitter. The authentication signature interference depends on the CBAT scheme utilized to embed the authentication signal into the message signal. Unfortunately, FEAT and Gelato were not designed to function correctly under such circumstances. However, our findings show that FREE can reliably authenticate transmitters even under such challenging conditions. This is the most distinguishing feature of FREE when compared to the prior art.

8.5.7 Blind Authentication

By design, all three schemes support blind authentication.

8.5.8 Security

Among the three schemes, FREE is the most robust scheme against interference as shown in Figure 8.4, it is also the most robust scheme against OOA jamming attack. Although FREE is prone to byzantine attack, we employ suitable mechanisms at the DFS to mitigate this attack as discussed in 8.4.2.

8.6 Experimental Validation

To evaluate the validity of FREE in a testbed environment, we implemented FREE on USRP radios. We used National Instruments' LabVIEW as the system-design platform to configure the USRP radios. We utilized the PHY-layer parameters shown in Table 8.1 to generate the message signal. We set the EFO as $f_a = 2.5$ kHz. In the experiments, we utilized USRP transmitters to transmit the embedded signals over the air, and USRP blind receivers to receive and extract the authentication signals. A PC was used as the DFS. The PC was also utilized to generate the time-stamps for recording the time-of-arrival of the reported data from the USRP blind receivers.

Figure 8.5 shows the LabVIEW block diagram of a blind receiver, illustrating the various steps needed to generate the decision variable. Recall that the decision variable is utilized by a blind receiver to detect one or more co-channel transmitters. In our experiments, we were able to verify that a blind receiver (using the implementation shown in Figure 8.5) is able to detect multiple co-channel transmitters that are simultaneously transmitting.

Figure 8.6 shows the performance of FREE in terms of BER vs. MSNR for the authentication signal. The figure includes BER curves generated from the LabVIEW implementation as well as those obtained from Matlab simulations. We can observe that the LabVIEW implementation's BER curves closely track those of the simulations, albeit the LabVIEW implementation's BER performance is slightly inferior to that of the simulations. This phenomenon can be attributed to the fact that time synchronization is assumed to be perfect in the simulations, but not in the testbed experiments.

8.7 Summary

In this paper, we proposed a novel CBAT scheme called FREE. Using theoretical analysis, simulations, and experimental results, we showed that FREE is the only PHY-layer au-

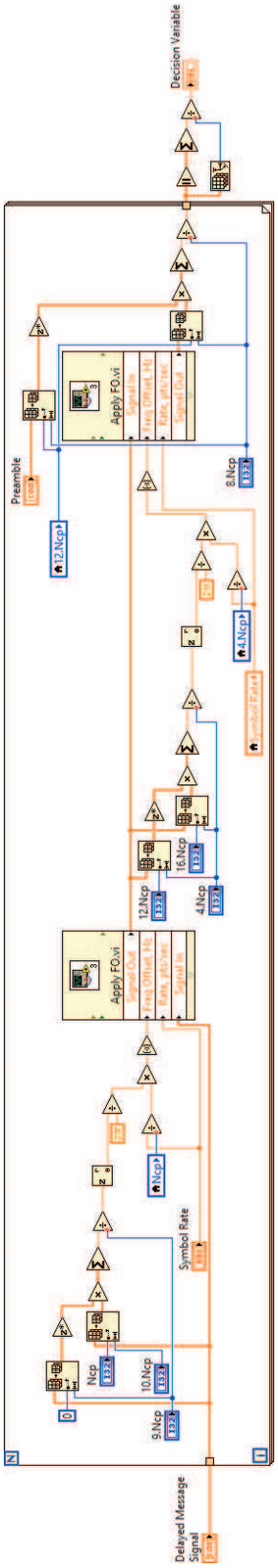


Figure 8.5: LabVIEW block diagram illustrating the implementation of FREE.

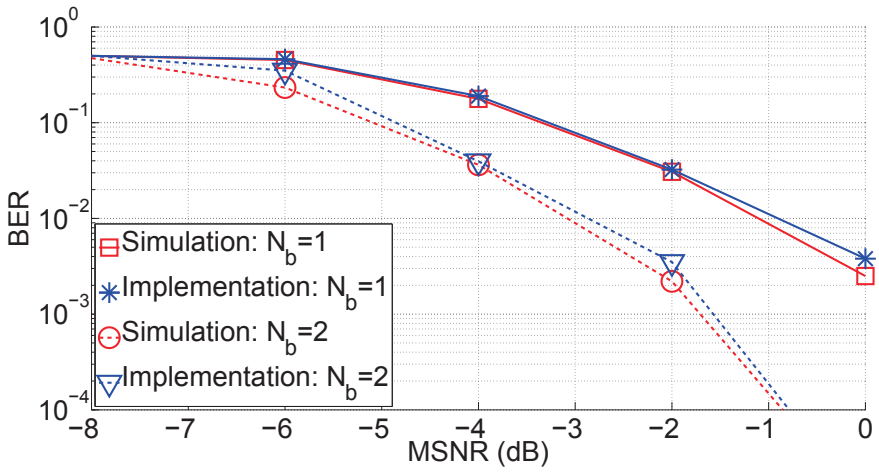


Figure 8.6: Comparison of the BER performance of the authentication signal obtained through simulation and implementation of FREE.

thentication scheme to date that can reliably authenticate multiple transmitters that are transmitting simultaneously in the same channel.

Chapter 9

Conclusion

In this dissertation, we discuss two primary technical problems in devising a transmitter authentication scheme for DSS: (1) how to generate and verify the authentication signal; and (2) how to embed and extract the authentication signal. For solving the first problem, we propose two privacy-preserving authentication (PPA) schemes. Firstly, we propose a novel group signature (GS) scheme called Group Signatures with Probabilistic Revocation (GSPR) which significantly reduces the computational complexity of the revocation check procedure compared to the GS's prior art. Secondly, we propose a novel direct anonymous attestation (DAA) scheme called Lightweight Anonymous attestation Scheme with Efficient Revocation (LASER), which significantly reduces the computational and communication complexity of the signature generation and verification procedures compared to the DAA's prior art.

After a thorough and comprehensive analysis of existing PPA schemes, the authors of [94] recently concluded that *revocation remains the major performance bottleneck of modern PPA schemes*, and that further research is urgently needed to design schemes offering better scalability with regard to revocation. In this dissertation, we proposed two novel schemes, GSPR and LASER, to tackle this problem. However, considering the fact that battery-powered radios with low computational capacity are employed in the DSS networks, they

may not be able to utilize any of the existing PPA schemes including GSPR and LASER, due to their high-computational cost of the on-line operations, i.e., signature generation and signature verification procedures. Hence, some of the ideas proposed in this dissertation, e.g., probabilistic revocation, user-controlled anonymity, tradeoff between off-line (i.e., obtaining keys/credentials) and on-line operations, etc., can be extended in future to design PPA schemes with low-computational cost in the on-line operations.

For solving the second problem, we propose three PHY-layer authentication schemes. Firstly, we propose an intended receiver-based authentication (IRA) scheme called Precoded Duobinary Signaling for Authentication (P-DSA), which does not suffer from the drawbacks of the blind signal superposition utilized in the prior art. Secondly, we propose a blind transmitter authentication (BTA) scheme called Frequency offset Embedding for Authenticating Transmitters (FEAT) which is the first scheme that satisfies all of the required criteria of an ideal BTA scheme. Thirdly, we propose a Crowd-Sourced Blind Authentication of Co-channel Transmitters (CBAT) scheme called FREquency offset Embedding for CBAT (FREE). According to our results obtained through analytical analysis, simulations, and experiments with an USRP-based implementation, FREE outperforms the existing PHY-layer authentication approaches, including P-DSA and FEAT, in all of the performance criteria that were considered.

Table 9.1 provides a qualitative comparison of the state of the art in PHY-layer authentication, including P-DSA, FEAT and FREE, in terms of the performance criteria discussed in Section 3.5. Note that FREE outperforms all other schemes in every respect except for authentication rate. Hence, designing a scheme which satisfies all the ideal attributes (discussed in Section 3.5) of a PHY-layer authentication scheme remains an open problem.

Table 9.1: Qualitative comparison of the PHY-layer authentication schemes based on the performance criteria.

Scheme	**Overhead**	**Compatibility**	**Authentication rate**	**Authentication signal's error performance**	**Authentication of concurrent transmission**	**Blind authentication**
[33]	Low	Good	Low	Medium	Good	Poor
[72]	Low	Poor	Low	Good	Good	Poor
[1]	High	Good	High	Poor	Poor	Poor
[31]	High	Poor	Low	Good	Poor	Medium
[29]	High	Good	High	Poor	Poor	Poor
P-DSA	Low	Poor	High	Medium	Poor	Poor
FEAT	Low	Good	Low	Good	Poor	Good
FREE	Low	Good	Low	Good	Good	Good

Bibliography

[1] X. Tan, K. Borle, W. Du, and B. Chen, "Cryptographic link signatures for spectrum usage authentication in cognitive radio," in *Proceedings of the Fourth ACM Conference on Wireless Network Security (WiSec)*, June 2011, pp. 79–90.

[2] P. Bichsel, J. Camenisch, G. Neven, N. P. Smart, and B. Warinschi, "Get shorty via group signatures without encryption," in *Security and Cryptography for Networks*. Springer Berlin Heidelberg, 2010, vol. 6280, pp. 381–398.

[3] E. Brickell and J. Li, "Enhanced privacy ID from bilinear pairing for hardware authentication and attestation," in *IEEE Second International Conference on Social Computing (SocialCom)*, 2010, pp. 768–775.

[4] D. Boneh and H. Shacham, "Group signatures with verifier-local revocation," in *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS)*, 2004, pp. 168–177.

[5] J. Camenisch, M. Drijvers, and A. Lehmann, "Anonymous attestation using the Strong Diffie Hellman assumption revisited," in *Proceedings of the 9th International Conference on Trust and Trustworthy Computing (TRUST)*, 2016, pp. 1–20.

[6] S. M. Dudley, W. C. Headley, M. Lichtman, E. Y. Imana, X. Ma, M. Abdelbar, A. Padaki, A. Ullah, M. M. Sohul, T. Yang, and J. H. Reed, "Practical issues for

spectrum management with cognitive radios," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 242–264, Mar. 2014.

[7] Federal Communications Commission, "Amendment of the commission's rules with regard to commercial operations in the 3550–3650 MHz band," *Report and Order and Second Further Notice of Proposed Rulemaking, GN Docket No 12–354*, April 2015.

[8] M. Altamimi, M. B. H. Weiss, and M. McHenry, "Enforcement and spectrum sharing: Case studies of federal-commercial sharing," *Available at SSRN 2310883*, Sept. 2013.

[9] J.-M. Park, J. H. Reed, A. A. L. Beex, T. C. Clancy, V. Kumar, and B. Bahrak, "Security and enforcement in spectrum sharing," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 270–281, Mar. 2014.

[10] C. Li, A. Raghunathan, and N. K. Jha, "An architecture for secure software defined radio," in *Design, Automation and Test in Europe (DATE)*, 2009, pp. 448–453.

[11] N. M. Smith, D. Johnston, G. W. Cox, and A. Shaliv, "Device, method, and system for secure trust anchor provisioning and protection using tamper-resistant hardware," 2012, US Patent Application 13/631, 562.

[12] S. Xiao, J.-M. Park, and Y. Ye, "Tamper resistance for software defined radio software," in *33rd IEEE International Computer Software and Applications Conference (COMPSAC)*, vol. 1, 2009, pp. 383–391.

[13] J. Y. Choi, M. Jakobsson, and S. Wetzel, "Balancing auditability and privacy in vehicular networks," in *Proceedings of the first ACM International Workshop on Quality of Service and Security in Wireless and Mobile Networks (Q2SWinet)*, 2005, pp. 79–87.

[14] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: A secure and privacy-preserving protocol for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3442–3456, Nov. 2007.

[15] E. Brickell, J. Camenisch, and L. Chen, "Direct anonymous attestation," in *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS)*, 2004, pp. 132–145.

[16] E. Brickell and J. Li, "Enhanced privacy ID: A direct anonymous attestation scheme with enhanced revocation capabilities," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 3, pp. 345–360, May 2012.

[17] H. Rahbari and M. Krunz, "Secrecy beyond encryption: obfuscating transmission signatures in wireless communications," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 54–60, Dec. 2015.

[18] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, Jan. 2007.

[19] H. Lu, J. Li, and M. Guizani, "A novel ID-based authentication framework with adaptive privacy preservation for VANETs," in *IEEE Computing, Communications and Applications Conference (ComComAp)*, Jan. 2012, pp. 345–350.

[20] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "An identity-based security system for user privacy in vehicular ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 9, pp. 1227–1239, Sept. 2010.

[21] T. Nakanishi and N. Funabiki, "A short verifier-local revocation group signature scheme with backward unlinkability," in *Advances in Information and Computer Security*. Springer Berlin Heidelberg, 2006, vol. 4266, pp. 17–32.

[22] Microsoft, "Device health attestation," https://technet.microsoft.com/en-us/library/mt750346.aspx, accessed: Nov 1, 2016.

[23] Google, "The chromium projects: TPM usage," http://www.chromium.org/developers/design-documents/tpm-usage, accessed: Nov 1, 2016.

[24] Trusted Computing Group, "TPM library specification," http://www.trustedcomputinggroup.org/tpm-library-specification/, accessed: Nov 1, 2016.

[25] ——, "TPM main specification," http://www.trustedcomputinggroup.org/tpm-main-specification/, accessed: Nov 1, 2016.

[26] L. Chen, D. Page, and N. P. Smart, "On the design and implementation of an efficient DAA scheme," in *International Conference on Smart Card Research and Advanced Application*, 2010, pp. 223–237.

[27] International Organization for Standardization, "ISO/IEC 11889-1:2009," http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html, accessed: Nov 1, 2016.

[28] L. Chen and J. Li, "Revocation of direct anonymous attestation," in *International Conference on Trusted Systems*, 2010, pp. 128–147.

[29] P. L. Yu, J. S. Baras, and B. M. Sadler, "Physical-layer authentication," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 38–51, Mar. 2008.

[30] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang, "Specguard: Spectrum misuse detection in dynamic spectrum access systems," in *IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 172–180.

[31] L. Yang, Z. Zhang, B. Y. Zhao, C. Kruegel, and H. Zheng, "Enforcing dynamic spectrum access with spectrum permits," in *Proceedings of the 13th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2012, pp. 195–204.

[32] X. Jin, J. Sun, R. Zhang, and Y. Zhang, "SafeDSA: Safeguard dynamic spectrum access against fake secondary users," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015, pp. 304–315.

[33] N. Goergen, T. C. Clancy, and T. R. Newman, "Physical layer authentication watermarks through synthetic channel emulation," in *IEEE Symposium on New Frontiers in Dynamic Spectrum*, Apr. 2010, pp. 1–7.

[34] T. Jiang, H. Zeng, Q. Yan, W. Lou, and Y. T. Hou, "On the limitation of embedding cryptographic signature for primary transmitter authentication," *IEEE Transactions on Wireless Communications Letters*, vol. 1, no. 4, pp. 324–327, Aug. 2012.

[35] S. Pasupathy, "Correlative coding: A bandwidth-efficient signaling scheme," *IEEE Communications Society Magazine*, vol. 15, no. 4, pp. 4–11, July 1977.

[36] V. Vadde and S. Gray, "Partial response signaling for enhanced spectral efficiency and RF performance in OFDM systems," in *IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 5, 2001, pp. 3120–3124.

[37] A. Dutta and M. Chiang, "See something, say something: Crowdsourced enforcement of spectrum policies," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 67–80, Jan. 2016.

[38] N. Kaufmann, T. Schulze, and D. J. Veit, "More than fun and money. Worker motivation in crowdsourcing - A study on mechanical turk," in *Proceedings of the 17th Americas Conference on Information Systems (AMCIS)*, 2011, Paper 340.

[39] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme," in *Advances in Cryptology - CRYPTO*. Springer Berlin Heidelberg, 2000, vol. 1880, pp. 255–270.

[40] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology - CRYPTO*, vol. 3152. Springer Berlin Heidelberg, 2004, pp. 41–55.

[41] C.-K. Chu, J. K. Liu, X. Huang, and J. Zhou, "Verifier-local revocation group signatures with time-bound keys," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, 2012, pp. 26–27.

[42] C.-I. Fan, R.-H. Hsu, and M. Manulis, "Group signature with constant revocation costs for signers and verifiers," in *Cryptology and Network Security*. Springer Berlin Heidelberg, 2011, vol. 7092, pp. 214–233.

[43] B. Libert, T. Peters, and M. Yung, "Group signatures with almost-for-free revocation," in *Advances in Cryptology - CRYPTO*. Springer Berlin Heidelberg, 2012, vol. 7417, pp. 571–589.

[44] T. Nakanishi, H. Fujii, Y. Hira, and N. Funabiki, "Revocable group signature schemes with constant costs for signing and verifying," in *Public Key Cryptography - PKC*. Springer Berlin Heidelberg, 2009, vol. 5443, pp. 463–480.

[45] E. Brickell and J. Li, "A pairing-based DAA scheme further reducing TPM resources," in *International Conference on Trust and Trustworthy Computing (TRUST)*, 2010, pp. 181–195.

[46] D. Bernhard, G. Fuchsbauer, E. Ghadafi, N. P. Smart, and B. Warinschi, "Anonymous attestation with user-controlled linkability," *International Journal of Information Security*, vol. 12, no. 3, pp. 219–249, 2013.

[47] E. Brickell, L. Chen, and J. Li, "A new direct anonymous attestation scheme from bilinear maps," in *International Conference on Trusted Computing*, 2008, pp. 166–178.

[48] ——, "Simplified security notions of direct anonymous attestation and a concrete scheme from pairings," *International Journal of Information Security*, vol. 8, no. 5, pp. 315–330, 2009.

[49] L. Chen, "A DAA scheme requiring less TPM resources," in *International Conference on Information Security and Cryptology*, 2009, pp. 350–365.

[50] L. Yang, J. Ma, Z. Liu, and R. Zheng, "A trusted authentication scheme for wireless networks using direct anonymous attestation," in *4th International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, 2012, pp. 279–285.

[51] L. Chen and R. Urian, "DAA-A: Direct anonymous attestation with attributes," in *International Conference on Trust and Trustworthy Computing (TRUST)*, 2015, pp. 228–245.

[52] L. Chen, Z. Cheng, and N. P. Smart, "Identity-based key agreement protocols from pairings," *International Journal of Information Security*, vol. 6, no. 4, pp. 213–241, 2007.

[53] C. H. Lim and P. J. Lee, "A key recovery attack on discrete log-based schemes using a prime order subgroup," in *Annual International Cryptology Conference*, 1997, pp. 249–263.

[54] J. Camenisch, M. Drijvers, and A. Lehmann, "Universally composable direct anonymous attestation," in *IACR International Workshop on Public Key Cryptography*, 2016, pp. 234–264.

[55] M. H. Au and A. Kapadia, "PERM: Practical reputation-based blacklisting without TTPs," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2012, pp. 929–940.

[56] L. Xi, J. Shao, K. Yang, and D. Feng, "ARBRA: Anonymous reputation-based revocation with efficient authentication," in *Proceedings of the 17th Information Security Conference (ISC)*, 2014, pp. 33–53.

[57] L. Chen and J. Li, "Flexible and scalable digital signatures in TPM 2.0," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2013, pp. 37–48.

[58] L. Xi, K. Yang, Z. Zhang, and D. Feng, "DAA-related APIs in TPM 2.0 revisited," in *International Conference on Trust and Trustworthy Computing (TRUST)*, 2014, pp. 1–18.

[59] L. Xi, D. Feng, Y. Qin, F. Wei, J. Shao, and B. Yang, "Direct anonymous attestation in practice: Implementation and efficient revocation," in *Twelfth Annual International Conference on Privacy, Security and Trust (PST)*, 2014, pp. 67–74.

[60] J. Hall, M. Barbeau, and E. Kranakis, "Detecting rogue devices in bluetooth networks using radio frequency fingerprinting," in *IASTED International Conference on Communications and Computer Networks*, Oct. 2006.

[61] W. Hou, X. Wang, and J.-Y. Chouinard, "Physical layer authentication in OFDM systems based on hypothesis testing of CFO estimates," in *IEEE International Conference on Communication (ICC)*, 2012, pp. 3559–3563.

[62] O. Ureten and N. Serinken, "Wireless security through RF fingerprinting," *Canadian Journal of Electrical and Computer Engineering*, vol. 32, no. 1, pp. 27–33, 2007.

[63] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2008, pp. 116–127.

[64] B. Danev and S. Capkun, "Transient-based identification of wireless sensor nodes," in *Proceedings of the IEEE International Conference on Information Processing in Sensor Networks*, Apr. 2009, pp. 25–36.

[65] K. A. Remley, C. A. Grosvenor, R. T. Johnk, D. R. Novotny, P. D. Hale, M. D. McKinley, A. Karygiannis, and E. Antonakakis, "Electromagnetic signatures of WLAN cards and network security," in *Proceedings of the Fifth IEEE International Symposium on Signal Processing Information Technology*, Dec. 2005, pp. 484–488.

[66] I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as communication with side information," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1127–1141, July 1999.

[67] C. Fei, D. Kundur, and R. H. Kwong, "Analysis and design of secure watermark-based authentication systems," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 1, pp. 43–55, Mar. 2006.

[68] J. E. Kleider, S. Gifford, S. Chuprun, and B. Fette, "Radio frequency watermarking for OFDM wireless networks," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, May 2004, pp. 397–400.

[69] L. Xiao, L. J. Greenstein, N. B. Mandayam, and W. Trappe, "Using the physical layer for wireless authentication in time-variant channels," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2571–2579, July 2008.

[70] X. Wang, Y. Wu, and B. Caron, "Transmitter identification using embedded pseudo random sequences," *IEEE Transactions on Broadcasting*, vol. 50, no. 3, pp. 244–252, Sept. 2004.

[71] Y. Liu, P. Ning, and H. Dai, "Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures," in *IEEE Symposium on Security and Privacy*, May 2010, pp. 286–301.

[72] R. Miller and W. Trappe, "Short paper: ACE: authenticating the channel estimation process in wireless communication systems," in *Proceedings of the Fourth ACM Conference on Wireless Network Security (WiSec)*, 2011, pp. 91–96.

[73] B. Danev, H. Luecken, S. Capkun, and K. El Defrawy, "Attacks on physical-layer identification," in *Proceeding of the 3rd ACM Conference on Wireless Network Security (WiSec)*, 2010, pp. 89–98.

[74] R. Chen, J. M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *The 27th IEEE Conference on Computer Communications (INFOCOM)*, April 2008, pp. 31–35.

[75] O. Fatemieh, R. Chandra, and C. A. Gunter, "Secure collaborative sensing for crowd sourcing spectrum data in white space networks," in *IEEE Symposium on New Frontiers in Dynamic Spectrum*, April 2010, pp. 1–12.

[76] R. Zhang, J. Zhang, Y. Zhang, and C. Zhang, "Secure crowdsourcing-based cooperative spectrum sensing," in *IEEE Conference on Computer Communications (INFOCOM)*, April 2013, pp. 2526–2534.

[77] S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for cryptographers," *Discrete Applied Mathematics*, vol. 156, no. 16, pp. 3113–3121, 2008.

[78] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 84, no. 5, pp. 1234–1243, 2001.

[79] M. J. Dworkin, "SHA-3 standard: Permutation-based hash and extendable-output functions," *Federal Information Processing Standards Publication (NIST FIPS)-202*, August 2015.

[80] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," in *Advances in Cryptology - CRYPTO*, 1997, pp. 410–424.

[81] J. Camenisch and V. Shoup, "Practical verifiable encryption and decryption of discrete logarithms," in *Advances in Cryptology - CRYPTO*, 2003, pp. 126–144.

[82] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances in Cryptology - CRYPTO*. Springer-Verlag, 1987, pp. 186–194.

[83] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[84] D. Boneh, "The decision diffie-hellman problem," in *Proceedings of the Third International Symposium on Algorithmic Number Theory*. Springer Berlin Heidelberg, 1998, vol. 1423, pp. 48–63.

[85] D. Boneh and X. Boyen, "Short signatures without random oracles and the SDH assumption in bilinear groups," *Journal of Cryptology*, vol. 21, no. 2, pp. 149–177, 2008.

[86] V. Goyal, "Reducing trust in the PKG in identity based cryptosystems," in *Advances in Cryptology - CRYPTO*. Springer Berlin Heidelberg, 2007, vol. 4622, pp. 430–447.

[87] C. Papamanthou, R. Tamassia, and N. Triandopoulos, "Optimal verification of operations on dynamic sets," in *Advances in Cryptology - CRYPTO*. Springer Berlin Heidelberg, 2011, pp. 91–110.

[88] R. Pickholtz, D. Schilling, and L. Milstein, "Theory of spread-spectrum communications-A tutorial," *IEEE Transactions on Communications*, vol. 30, no. 5, pp. 855–884, May 1982.

[89] M. Bellare, D. Micciancio, and B. Warinschi, "Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions," in *Advances in Cryptology - EUROCRYPT*. Springer Berlin Heidelberg, 2003, vol. 2656, pp. 614–629.

[90] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith, "PEREA: Towards practical TTP-free revocation in anonymous authentication," in *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS)*, 2008, pp. 333–344.

[91] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology*, pp. 361–396, 2000.

[92] M. O. Rabin, "Probabilistic algorithm for testing primality," *Journal of Number Theory*, vol. 12, no. 1, pp. 128–138, 1980.

[93] E. H. Dinan and B. Jabbari, "Spreading codes for direct sequence CDMA and wideband CDMA cellular networks," *IEEE Communications Magazine*, vol. 36, no. 9, pp. 48–54, Sept. 1998.

[94] M. Manulis, N. Fleischhacker, F. Gunther, F. Kiefer, and B. Poettering, "Group signatures - authentication with privacy," Group Signatures Study for BSI - German Federal Office for Information Security, Tech. Rep., 2012.

[95] "PBC: Pairing-based cryptography," https://crypto.stanford.edu/pbc/, accessed: May 1, 2016.

[96] P. S. L. M. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," in *International Workshop on Selected Areas in Cryptography*, 2005, pp. 319–331.

[97] Trusted Computing Group, "Algorithm registry: Revision 01.22," https://www.trustedcomputinggroup.org/wp-content/uploads/TCG_Algorithm_Registry_Rev_1.22.pdf, accessed: Nov 1, 2016.

[98] K. Goldman, "IBM TPM 2.0 TSS," http://ibmswtpm.sourceforge.net/ibmtss2.html, accessed: Nov 1, 2016.

[99] ——, "IBM TPM 2.0 emulator," http://ibmswtpm.sourceforge.net/ibmswtpm2.html, accessed: Nov 1, 2016.

[100] B. Yang, D. Feng, and Y. Qin, "A lightweight anonymous mobile shopping scheme based on DAA for trusted mobile platform," in *IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, 2014, pp. 9–17.

[101] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. McGraw-Hill, 2008.

[102] G. D. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Transactions on Information Theory*, vol. 18, no. 3, pp. 363–378, May 1972.

[103] ——, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[104] P. L. Yu, J. S. Baras, and B. M. Sadler, "Multicarrier authentication at the physical layer," in *International Symposium on World of Wireless, Mobile and Multimedia Networks*, June 2008, pp. 1–6.

[105] J.-S. Um, S.-H. Hwang, and B.-J. Jeong, "A comparison of PHY layer on the Ecma-392 and IEEE 802.11af standards," in *Seventh International ICST Conf. on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, 2012, pp. 313–319.

[106] "Design considerations for minimum SNR," http://www.cisco.com/en/US/docs/wireless/technology/mesh/7.3/design/guide/Mesh_chapter_011.pdf, accessed: May 15, 2016.

[107] T. Yucek and H. Arslan, "OFDM signal identification and transmission parameter estimation for cognitive radio applications," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2007, pp. 4056–4060.

[108] C. R. N. Athaudage and K. Sathananthan, "Cramer-Rao lower bound on frequency offset estimation error in OFDM systems with timing error feedback compensation," in *Fifth IEEE International Conference on Information, Communications and Signal Processing*, 2005, pp. 1231–1235.

[109] M.-H. Cheng and C.-C. Chou, "Maximum-likelihood estimation of frequency and time offsets in OFDM systems with multiple sets of identical data," *IEEE Transactions on Signal Processing*, vol. 54, no. 7, pp. 2848–2852, July 2006.

[110] "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," *IEEE Standard 802.11-2012*.

[111] K. Lauter, "The advantages of elliptic curve cryptography for wireless security," *IEEE Wireless Communications Magazine*, vol. 11, no. 1, pp. 62–67, 2004.

[112] T. Fusco and M. Tanda, "Blind synchronization for OFDM systems in multipath channels," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1340–1348, 2009.

[113] H. Ishii and G. W. Wornell, "OFDM blind parameter identification in cognitive radios," in *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 1, Sept. 2005, pp. 700–705.

[114] A. Punchihewa, V. K. Bhargava, and C. Despins, "Blind estimation of OFDM parameters in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 10, no. 3, pp. 733–738, Mar. 2011.

[115] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang, "Fine-grained channel access in wireless LAN," in *Proceedings of the ACM SIGCOMM Conference*, 2010, pp. 147–158.

[116] D. Bladsj, M. Hogan, and S. Ruffini, "Synchronization aspects in LTE small cells," *IEEE Communications Magazine*, vol. 51, no. 9, pp. 70–77, Sept. 2013.

[117] G. Ren, H. Zhang, and Y. Chang, "SNR estimation algorithm based on the preamble for OFDM systems in frequency selective channels," *IEEE Transactions on Communications*, vol. 57, no. 8, pp. 2230–2234, Aug 2009.

[118] A. F. Molisch and M. Z. Win, "MIMO systems with antenna selection," *IEEE Microwave Magazine*, vol. 5, no. 1, pp. 46–56, Mar. 2004.