SECURITY, PRIVACY AND TRUST IN WIRELESS MESH NETWORKS

by
AHMET ONUR DURAHİM

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Doctorate of Philosophy

Sabancı University
March, 2012

SECURITY, PRIVACY AND TRUST IN WIRELESS MESH NETWORKS

APPROVED BY

Assoc. Prof. Erkay Savaş .....
(Thesis Supervisor)

Assoc. Prof. Albert Levi .....

Assoc. Prof. Cem Güneri .....

Asst. Prof. Selim Balcısoy .....

Asst. Prof. Selçuk Baktır .....

DATE OF APPROVAL: .....

March, 2012

SECURITY, PRIVACY AND TRUST IN WIRELESS MESH NETWORKS

Ahmet Onur Durahim

CSE, PhD Thesis, 2012

Thesis Supervisor: Assoc. Prof. Erkay Savaş

Keywords: Network Security, Wireless Mesh Networks, Privacy-aware Authentication, Accountability, Group Signatures

Abstract

With the advent of public key cryptography, digital signature schemes have been extensively studied in order to minimize the signature sizes and to accelerate their execution while providing necessary security properties. Due to the privacy concerns pertaining to the usage of digital signatures in authentication schemes, privacy-preserving signature schemes, which provide anonymity of the signer, have attracted substantial interest in research community.

Group signature algorithms, where a group member is able to sign on behalf of the group anonymously, play an important role in many privacy-preserving authentication/identification schemes. On the other hand, a safeguard is needed to hold users accountable for malicious behavior. To this end, a designated opening/revocation manager is introduced to open a given anonymous signature to reveal the identity of the user. If the identified user is indeed responsible for malicious activities, then s/he can also be revoked by the same entity. A related scheme named direct anonymous attestation is proposed for attesting the legitimacy of a trusted computing platform while maintaining its privacy.

This dissertation studies the group signature and direct anonymous attestation schemes and their application to wireless mesh networks comprising resource-constrained embedded devices that are required to communicate securely and be authenticated anonymously, while malicious behavior needs to be traced to its origin. Privacy-aware devices that anonymously connect to wireless mesh networks also need to secure their communication via efficient symmetric key cryptography, as well.

In this dissertation, we propose an efficient, anonymous and accountable mutual authentication and key agreement protocol applicable to wireless mesh networks. The proposed scheme can easily be adapted to other wireless networks. The proposed scheme is implemented and simulated using cryptographic libraries and simulators that are widely deployed in academic circles. The implementation and simulation results demonstrate that the proposed scheme is effective, efficient and feasible in the context of hybrid wireless mesh networks, where users can also act as relaying agents.

The primary contribution of this thesis is a novel privacy-preserving anonymous authentication scheme consisting of a set of protocols designed to reconcile user privacy and accountability in an efficient and scalable manner in the same framework. The three-party join protocol, where a user can connect anonymously to the wireless mesh network with the help of two semi-trusted parties (comprising the network operator and a third party), is efficient and easily applicable in wireless networks settings. Furthermore, two other protocols, namely two-party identification and revocation protocols enable the network operator, with the help of the semi-trusted third party, to trace suspected malicious behavior back to its origins and revoke users when necessary. The last two protocols can only be executed when the two semi-trusted parties cooperate to provide accountability. Therefore, the scheme is protected against an omni-present authority (e.g. network operator) violating the privacy of network users at will. We also provide arguments and discussions for security and privacy of the proposed scheme.

ÇOKGEN BAĞLANTILI KABLOSUZ AĞLARDA GÜVENLİK, MAHREMİYET, VE GÜVEN

Ahmet Onur Durahim

CSE, Doktora Tezi, 2012

Tez Danışmanı: Doç. Dr. Erkay Savaş

Anahtar Kelimeler: Ağ Güvenliği, Çokgen Bağlantılı Kablosuz Ağlar, Mahremiyet-bilinçli doğrulama, Sorumlu tutulabilirlik, Grup imzaları

Özet

Açık anahtarlı şifrelemenin gelişmesiyle, gerekli güvenlik özelliklerini sağlayarak, imza boyutlarını mümkün olduğu kadar küçültmek ve çalışmalarını hızlandırmak amacıyla sayısal imza düzenleri kapsamlı olarak çalışılmıştır. Sayısal imzaların doğrulama düzenlerindeki kullanımından dolayı ortaya çıkan mahremiyet endişesinden dolayı, imza atan kişilerin gerçek kimliğini saklayan mahremiyet-koruyucu imza düzenleri araştırma topluluğunda büyük ilgi çekmiştir.

Herhangi bir grup üyesinin bilinmeden grup adına imza atabildiği Grup imza algoritmaları, mahremiyet-koruyucu doğrulama/tanılama düzenlerinde önemli bir rol oynamaktadırlar. Diğer taraftan, kullanıcıları kötü niyetli davranışlarından sorumlu tutmak için önlem almak gerekmektedir. Bu amaçla, eldeki anonim imzayı açarak, bu imzayı atan kullanıcının kimliğini ortaya çıkarması için belirlenmiş açan (iptal eden) yönetici tanımlanmıştır. Kimliği ortaya çıkartılan kullanıcı, kötü niyetli davranışlarin sorumlusu ise, bu kullanıcı kimliğini ortaya çıkaran varlık tarafından ağdan menedilebilir. Bununla

ilişkili olarak, güvenilir bilişim platformunun mahremiyetini koruyarak meşruiyetini tasdik etmesini sağlayan direk anonim tasdik adı verilen düzen önerilmiştir.

Bu tezde öncelikle önerilmiş grup imzalari ve direk anonim tasdik düzenleri incelenmiştir. Analiz edildikten sonra bu düzenler, güvenli iletişim kurmaları ve anonim olarak doğrulanmaları gereken kaynak-kısıtlı gömülü cihazlardan oluşan çokgen bağlantılı kablosuz ağlara uyarlanmıştır. Bunlar sağlanırken, kötü niyetli davranışların da kaynağına kadar izlenebilmeleri gerekmektedir. Ayrıca, ağa anonim bağlanmaları gereken mahremiyetlerinin farkındaki cihazların iletişimlerini çok daha verimli olan gizli anahtarlı şifreleme ile korumaları gerekmektedir.

Bu tezde, çokgen bağlantılı kablosuz ağlara uygulanabilir, verimli, anonim ve aynı zamanda sorumlu tutulabilir karşılıklı doğrulama ve anahtar anlaşma protokolü önerilmiştir. Önerilen düzen diğer kablosuz ağlara da kolayca uyarlanabilmektedir. Önerilen düzen, akademik çevrelerde yaygın olarak kullanılan kripto kütüphanelerini ve benzetimcilerini kullanarak uygulanmış ve benzetimleri yapılmıştır. Bu uygulama ve benzetim sonuçları, önerilen düzenin, kullanıcıların aynı zamanda yönlendirici görevinde de bulunabildiği melez çokgen bağlantılı kablosuz ağlar bağlamında etkili, verimli ve uygulanabilir olduğunu göstermektedir.

Bu tezin ana katkısı, kullanıcı mahremiyetini ve sorumlu tutulabilirliğini verimli ve ölçeklenebilir bir şekilde aynı çerçevede uzlaştırmak için tasarlanmış protokollerden oluşan yeni mahremiyet-koruyucu anonim doğrulama düzenidir. Kullanıcının, bir ağ operatörü ve bir üçüncü taraftan oluşan iki yarı-güvenilir tarafın yardımıyla, anonim olarak çokgen bağlantılı kablosuz ağa bağlanabildiği üç-taraflı katılım protokolü, kablosuz ağlara kolay ve verimli bir şekilde uygulanabilmektedir. Ayrıca, iki-taraflı tanımlama ve feshetme adı verilen diğer iki protokol ile ağ operatörü, yarı-güvenilir üçüncü tarafın yardımıyla, şüphelenilen kötü niyetli davranışları çıkış noktasına kadar izleyip, gerekli gördüğünde kullanıcıları ağdan menedebilmektedir. Bahsi geçen son iki protokol, sorumlu tutulabilirliği sadece iki yarı-güvenilir tarafın işbirliği ile sağlayabilmektedir. Böylece, düzen, istediğinde ağ kullanıcılarının mahremiyetini ihlal eden heryerde bulunabilen yetkiliye (örneğin, ağ operatörü) karşı korunmaktadır.

to my beloved family, brothers & sisters

Acknowledgments

I am really grateful to Prof. Erkay Savaş for his support and guidance starting from the beginning of my PhD journey to the end. I feel privileged for working under his supervision. Without his guidance and valuable advices, my PhD would not come to an end.

I am greatly indebted to Prof. Albert Levi for his valuable advices and the discussions made during the course of my PhD. I have learned much from him.

I would like to thank TÜBİTAK (The Scientific and Technical Research Council of Turkey) for his support under Project Number 105E089 (TUBITAK Career Award).

I would also like to thank my jury members, Prof. Cem Güneri, Prof. Selim Balcısoy and Prof. Selçuk Baktır for their valuable review and comments on this dissertation.

I would like to give special thanks to İsmail Fatih Yıldırım for being a generous friend to me and for his help in the development and coding of the simulations. I am grateful to Ömer Bakkalbaşı for the proof reading of this thesis. I would also like to thank Prof. Ali Rana Atılgan for his support and guiding discussions.

Finally, I would like to thank my parents, Hamdi, Nilgün and Seçkin Durahim, for their patience and support during my long lasting PhD life.

Table of Contents

Abstract **iv**

Özet **vi**

Acknowledgments **ix**

1 Introduction **1**
1.1 Wireless Mesh Networks 4
1.2 Security and Privacy Requirements for Wireless Mesh Networks 6
1.3 Motivation and Contributions 7
1.3.1 Contributions 11
1.4 Summary of the Thesis 12

2 Foundations and Basic Protocols **15**
2.1 Notations and Preliminaries 15
2.2 Number Theoretic Assumptions 16
2.3 Signature Proof of Knowledge 19

3 Elliptic Curve and Pairing Based Cryptography **26**
3.1 Elliptic Curve Cryptography 26
3.1.1 Elliptic Curves over Finite Fields 26
3.1.2 Elliptic Curve Cryptosystems 29
3.1.3 Attacks on Elliptic Curves 30
3.2 Pairing Based Cryptography 32
3.2.1 Bilinear Pairings 33
3.2.2 Hardness Assumptions in Pairing-based Cryptography 35
3.2.3 Pairing Implementations 37
3.2.4 Pairing-friendly Curves 39
3.2.4.1 Supersingular Elliptic Curves 40
3.2.4.2 Ordinary Curves 42

4 Group Signatures and Attestation Schemes **46**
4.1 Introduction to Group Signatures 46
4.2 Properties of the Group Signature Schemes 47
4.3 Evolution of Group Signatures 50

4.3.1 Group Signature Approach of Camenisch and Stadler [1] 53
4.3.2 Provably Secure Group Signatures against Coalition Attacks . . . 55
4.4 Revocation in Group Signatures 57
4.5 Pairing based Group Signatures 61
4.6 Direct Anonymous Attestation 65

5 A^2-MAKE: Anonymous and Accountable Authentication Framework for Wireless Mesh Networks 69
5.1 Introduction 69
5.1.1 Introduction and Motivation 70
5.1.2 Related Work 72
5.2 Network Architecture and Problem Formulation 75
5.3 Our Construction 78
5.3.1 Setup 78
5.3.2 Join Protocol 79
5.3.3 MAKE - Mutual Authentication and Key agrEement Protocol . . 81
5.4 User Accountability and Key Revocation 86
5.4.1 Identify - (User identification without private key extraction) . . 88
5.4.2 Revoke - (User revocation with private key extraction) 89
5.5 Security and Performance Analysis 89
5.5.1 Security Analysis 89
5.5.2 Performance Analysis 97
5.5.2.1 Computational Overhead 97
5.5.2.2 Communication Overhead 99
5.6 Implementation and Timing Analysis 101
5.6.1 Timing Results for a Resource Constrained User 104
5.7 Simulation Results 104
5.7.1 Scenario 1: UserRL is held both at mesh routers and mesh clients 108
5.7.2 Scenario 2: UserRL is held only at mesh routers 113

6 Concluding Remarks 117

Bibliography 120

List of Figures

1.1 Hybrid WMN architecture 5

5.1 Join Protocol: Generation of Group Secret Keys and Associated Credentials 79
5.2 Authentication Times at 80-bit Security Level 108
5.3 Authentication Times at 128-bit Security Level 109
5.4 Number of Successful Authentications by Routers and Relaying Agents . 110
5.5 Ratio of Successful Authentication Attempts (Weighted average of Relaying agent and Router Authentications) 111
5.6 Ratio of Successful Authentication Attempts (Relaying agent and Router Authentications are shown separately) 111
5.7 True Positive Authentications made by Relaying Mesh Clients 112
5.8 Authentication Times at 80-bit Security Level 113
5.9 Authentication Times at 128-bit Security Level 114
5.10 Number of Successful Authentications by Routers and Relaying Agents . 115
5.11 Ratio of Successful Authentication Attempts (Weighted average of Relaying agent and Router Authentications) 115
5.12 Ratio of Successful Authentication Attempts (Relaying agent and Router Authentications are shown separately) 116

List of Tables

3.1 Supersingular curves and their Distortion maps,(*embedding degree, security multiplier) 43
3.2 Characterization of ordinary elliptic curves due to Miyaji et al. [2] 44

4.1 Comparison of Pairing based Group Signature Schemes 64
4.2 Complexity and assumptions of the scheme of [3] 66

5.1 Computational Overhead of A^2-MAKE and PEACE [4] 98
5.2 Communication Overhead of A^2-MAKE (*optional) 100
5.3 Comparison of the Communication Overhead (Signature Sizes) 100
5.4 Timing Results of the 160-bit Implementation of A^2-MAKE 102
5.5 Timing Results of the 256-bit Implementation of A^2-MAKE 103
5.6 Time Costs of UserRL Checking for 1, 10, 50, 100 and 200 Rogue Users . 104
5.7 Detailed Timings for the Protocol Steps taken by the Network User 105
5.8 Detailed Timings for the Protocol Steps executed by the Network User on an Embedded Processor 105

Chapter 1

Introduction

Cryptography, meaning secret writing, is the science of delivering critical information securely over insecure communication channels. Security can be obtained so that messages that are being eavesdropped cannot be understood by an adversary (confidentiality), that their content cannot be changed by unauthorized parties without being detected (integrity), and that each communicating party is ensured that it is talking to the intended entity (authentication).

Cryptography was initially used largely for military purposes to secure critical information that can be overheard by enemies. In early years, cryptography was solely based on the symmetric techniques where communicating parties share a common key for cryptographic usage, i.e. same key is used for both encrypting and decrypting messages. In the digital world, symmetric key cryptography can be used to provide confidentiality via encryption and integrity via message authentication codes. However, it does not provide the means for undeniable digital signatures which form a binding between the user and message formed/delivered by the user. Non-repudiation property of digital signatures, which is the ability to ensure that a party cannot deny that she is the originator of a digital signature actually generated by herself for a message/document, is also a requirement for the digital signatures to replace the handwritten signatures used in critical communications and documents, such as legal commercial agreements.

Another important drawback of symmetric key cryptography is the requirement for pre-existence of a shared secret key between communicating parties. This requirement thus necessitates means for secure key distribution. Therefore, constructing a secure

channel for distributing secret keys among communicating parties efficiently is of critical importance. Without the means for distributing keys, communicating parties must either agree on secret keys by meeting in person or through a trustworthy carrier.

A breakthrough in the history of cryptography was achieved by Diffie and Hellman [5] in their seminal paper "New Directions in Cryptography", whereby they introduced the concept of public-key cryptography, which makes undeniable digital signatures and key exchange possible without the need to share keys a priori. In public key cryptography, each user possesses two different keys related in a number theoretic way, one of which is private and only known by the user himself and the other one is publicly known by everyone with a proof that binds the key to its owner. So, one uses the other party's public key, for example, to encrypt a given message and obtain resultant ciphertext which can only be decrypted by the corresponding private key known only by the intended party. In their paper, authors proposed the first key exchange protocol widely known as Diffie-Hellman key exchange.

Subsequently, other public key cryptosystems are proposed such as RSA cryptosystem by Rivest et al. [6] and ElGamal cryptosystem by El Gamal [7], along with their corresponding digital signature schemes. Digital signatures are then formalized by Goldwasser et al. [8]. Following the invention of digital signatures, authentication mechanisms are developed utilizing the proposed digital signature schemes. This, in turn, created privacy concerns in certain applications due to the fact that one is implicitly identified uniquely by her digital signature. As a result, in order to avoid privacy problems, various approaches have been proposed for anonymous authentication of privacy-aware users, such as group signatures [9, 10, 11] and ring signatures [12, 13].

In group signature schemes, members of a certain group can sign messages (documents) on behalf of the group anonymously. This way, one may acquire credentials which prove that the owner is eligible to obtain services that are provided only to that certain group. However, anonymity brings about accountability issues: malicious users with anonymous authentication need to be identified later and thus held responsible for their possible malevolent actions. Therefore, in order to prevent such issues, a designated entity called group manager is empowered with the capability of opening signatures to

reveal the identities of signers when needed. But, this also means a potential compromise of the user privacy by this powerful entity. Therefore, there is a trade-off between providing anonymity and accountability which have conflicting goals; the former is trying to hide the identity of the user, while the latter is trying to reveal it.

In this thesis, we address the issue of reconciling these conflicting objectives within a practical authentication framework that also incorporates a key agreement scheme to secure the communication between the user being authenticated and the corresponding verifier. We devise a set of efficient protocols, constituting the framework, specifically for hybrid wireless mesh networks where the ad hoc nature of the network and resource constraints of user devices pose complex and multi-faceted challenges. First of all, we correctly identify the security, privacy and trust challenges in wireless mesh (or similar) networks. While users of such networks should be protected against the adversaries or other third parties, we cannot let them be susceptible to arbitrary intervention and/or tracking by an omni-present and omni-potent network operator, advantageously situated with respect to other users. We, therefore, have to protect the privacy of network users against the network operator as well, which is in fact one of the most challenging tasks in such networks. On the other hand, absolute privacy without any fallback mechanism can lead to some irresponsible and malicious user behaviour which cannot be traced back to its origin. However, the right of executing a mechanism for identifying such users should be distributed between the network operator and a trusted third party which will act justly and impartially.

The most important aspects of the solution are that it must be lightweight on user side while scalable on the sides of network operator and the trusted third party. The use of *fully* trusted parties is infeasible and render the solution inapplicable in real usage scenarios where a party that enjoys the full trust by all parties is impractical to implement. Therefore, we relax the trust requirements on the third party to a degree that existing solutions such as certificate authorities can be used as a model to design such third parties.

The proposed model in this thesis achieves these requirements in an efficient and practical manner while creating a reciprocal trust relationship between the users and the network operator. The implementation and simulation results of the proposed framework

demonstrate its suitability on hybrid wireless mesh (or many other ad hoc) networks. The proposed framework provides an efficient, accountable, and at the same time, privacy-preserving authentication and key agreement mechanism for wireless mesh networks consisting of resource-constrained embedded devices, whereby legitimate users can connect to the network (and obtain provided services) from anywhere without being identified or tracked arbitrarily.

1.1 Wireless Mesh Networks

Nowadays, wireless mesh networks (WMNs) emerge as a promising technology to provide low cost and scalable solutions for high speed Internet access and additional services. Thus, it is no surprise that it has been the focus of increasing attention of all quarters from research community to industry and military.

A WMN is a dynamically self-organized and self-configured network, where the nodes automatically establish and maintain mesh connectivity in a collaborative fashion. The collaborative nature of the mesh networks results in low up-front cost, easy network maintenance, robustness and reliable service coverage [14]. In their simplest form, WMNs are comprised of mesh routers and mesh clients (network users), whereby mesh routers are in charge of providing coverage and routing services for mesh clients which connect to the networks using laptops, PDAs, smartphones, etc. Hybrid architectures [14] (*cf.* Figure 1.1) are the most popular since in addition to mesh routers, mesh users may also perform routing and configuration functionalities for other users to help improve the connectivity and coverage of the network. In other words, any node in the network can act both as a router and as a user resulting in hybrid architectures.

In order to ensure wide user-acceptance and deployment of WMNs, *security* and *privacy* concerns of users need to be addressed in an efficient and reliable manner. Due to the dynamic and open nature of the network, it is essential to provide effective access control mechanisms to guarantee the registered users a reliable network connectivity and other security services for the protection of network communication. On one hand, user

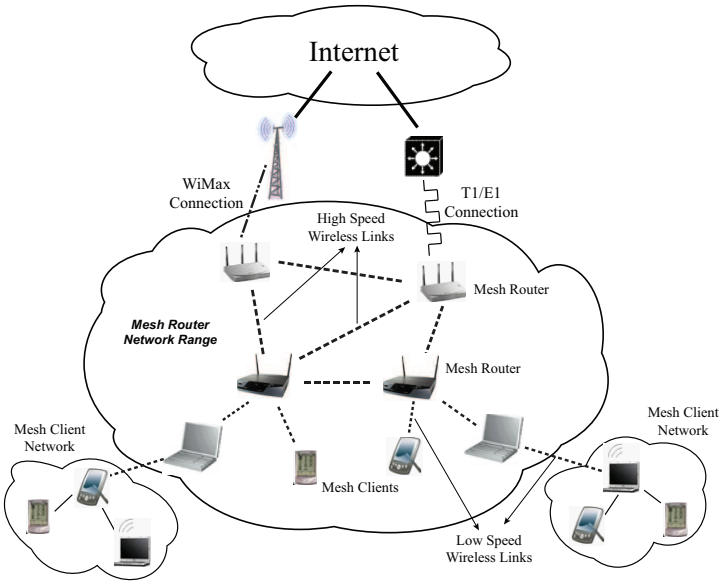

Figure 1.1: Hybrid WMN architecture

privacy is needed during authenticated connection to the network. On the other hand, user *accountability* is required in order to detect misbehaving users and, if needed, deny network access to them via revoking. Therefore, access control, security, user privacy and accountability objectives can conflict with each other, making it difficult to reconcile within the same framework.

Hybrid wireless mesh networks require that resource and energy constrained mesh clients perform costly operations necessary to provide relaying. The proposed security architecture treats performance and energy usage as extremely crucial issues. Therefore, the main requirements for a security framework that is to be accepted and widely deployed involve efficient signature generation and verification mechanisms (utilized in anonymous authentication) employing smaller key sizes as well as efficient key sharing and other security operations with minimal communication. If one wants to provide access control via anonymous authentication together with confidentiality and/or integrity, then an efficient key agreement scheme should be incorporated into the proposed authentication scheme. This way, existing efficient symmetric key cryptographic algorithms can be used to secure the communication of authorized users. It is important to note that, a trade-off between

efficiency and either of security and/or privacy should be avoided. Any improvement made on the performance of the proposed scheme that entails a reduction in security and privacy requirements is unacceptable.

Therefore, the most challenging requirement for WMNs is the design of an access control mechanism that provides both anonymous authentication to its privacy-aware users who should also be held accountable for their malicious activities. Besides, efficient secure communication between the network user and authenticating mesh router should also be provided via symmetric key sharing for the framework to be widely acceptable for practical usage.

1.2 Security and Privacy Requirements for Wireless Mesh Networks

The following security requirements are the objectives that need to be efficiently achieved in an anonymous and accountable authentication framework proposed for the wireless mesh networks;

1. ***Confidentiality/Integrity***: Efficient symmetric key establishment protocol is required where both sender and the recipient share a key for protecting communications between a mesh client and a mesh router (or a relaying mesh client). This is achieved via symmetric key encryption and message authentication codes.

2. ***Authentication***: Authentication is required to be performed anonymously by legitimate users to connect to the network (and to obtain required services).

3. ***User Privacy***: User privacy is achieved if the framework provides anonymity and unlinkability at the same time. As users authenticate themselves using signature-based schemes, the following signature properties are needed for these requirements[1];

[1] User-Controlled Linkability is an optional requirement.

a. *Anonymity*: Given a valid signature, identifying the signer (i.e. owner of the signature) must be computationally hard [10, 11].

b. *Unlinkability*: Given a list of signatures, where some of them are generated by the same user, no other party can link any two of the valid signatures generated by the same authorized user [10, 11]. Even, no one is able to determine whether any two of these valid signatures are generated by different users or by the same one.

c. *User-Controlled Linkability*: In certain situations, a user may want to be tracked for a given period of time without being identified. In addition, an authenticator may also enforce tracking of users in order to prevent anonymity-based attacks such as Sybil attacks [15]. To achieve this, the user and the authenticator can devise a scheme, under which the latter can link signatures generated by the user for a period of time determined by the former. The scheme compromises neither the identity of the user nor her private key.

4. ***User Accountability and Revocation***: Users should be held accountable for their actions. When they are involved in unacceptable and destructive activities, they need to be identified, and even revoked if necessary. Thus, anonymity and unlinkability properties are relaxed against a specific authority usually known as *the opener/revocation manager*, which acquires the right to identify and/or revoke users when certain conditions are met.

1.3 Motivation and Contributions

As seen from the previous discussions, an anonymous and accountable authentication framework which incorporates a key agreement scheme should satisfy the security and privacy requirements mentioned in the previous section in an efficient manner. The hybrid wireless mesh networks require an efficient solution from both computational and communication perspectives. To the best of our knowledge, none of the previously proposed solutions satisfactorily fulfilled all the security and privacy requirements in an efficient

manner.

Furthermore, network and/or service providers may need user-controlled linkability of network users[2] to prevent anonymity based attacks and/or to design a pricing structure for the provided services.

In order to provide an efficient and acceptably secure solution, first we analyzed the group signatures schemes, specifically an advanced application of group signatures known as direct anonymous attestation schemes. User-controlled linkability along with the efficiency requirements lead us to the efficient direct anonymous attestation proposal of Chen et al. [16] that additionally provides optional user-controlled linkability which is not addressed by the existing group signature schemes in literature. The scheme by Chen et al. [16] forms the basis of signature generation and verification protocols used in our proposed framework due to its small signature size and efficient signature generation and verification algorithms.

Moreover, it is important to separate the identification and revocation mechanisms in order to provide accountability that is acceptable from user privacy perspective. Accountability requirement can be incorporated into the authentication scheme in conjunction with a suitable *join* protocol, which is executed when user is initiated to the network. Since the network operator deploys all the mesh routers in our construction and forms a well-connected network (thus being the most powerful entity within the network), it should not have access to secret signing keys of mesh clients as proposed by Ren and Lou [4]. Doing so will violate the unlinkability property of the generated signatures and empowering the network operator as the sole party that can identify and revoke any user by itself. On the other hand, because the mesh clients are registered to the network operator and network operator is highly accessible and the first to detect any malicious behavior, it is necessary to involve it in identification and revocation protocols. In this respect, we device a join protocol and corresponding protocols that provide accountability in a way that no single authority is able to perform the identification and revocation of mesh network clients. In the proposed scheme this right is entrusted to the network operator together

[2] In order to accomplish this, router and the mesh client together decide on a session basename which provides linkability of the signatures generated under the same basename.

with a trusted third party. One cannot exercise this right without the participation of the other.

Certificate Revocation List (CRL) based (cf. Section 4.4 - a) revocation mechanism is adopted into the framework which fits best in our construction. We named this list as UserRL, an abbreviation for the user revocation list. Users are revoked by a two-party revocation protocol which adds the secret signing key of the malicious user into the UserRL. Revoked users are prevented from accessing the network services if the signature used in anonymous authentication is originated from a user whose secret signing key is included in UserRL. However, before revoking access rights of a suspicious user, she must be identified first. The identification algorithm should not reveal the secret signing key of the user in question. If the user is convicted of destructive malicious activities, then the revocation procedure should be performed. In order to achieve these operations separately and independently, identification of a suspected user and revocation of malicious users are performed with two different protocols.

In the proposed framework, parties that comprise the hybrid mesh network are the network operator (NO), a semi-trusted third party (STTP)[3], a number of routers and a number of mesh clients (also mentioned as network users).

In the following, we describe the approach used to provide the security and privacy requirements mentioned previously;

- ***Confidentiality and Integrity :*** Communications are secured by efficient symmetric key algorithms which require communicating parties to pre-share symmetric secret keys. In our proposal, an authenticated Diffie-Hellman key exchange procedure is incorporated into the anonymous authentication scheme to establish a symmetric key between network user and a relaying agent, either a router or another network user. This key only secures the communication between the parties performing the proposed mutual authentication procedures. In every session that is successfully established via anonymous authentication, a new secret session key is formed making use of random nonces. This way, even if an attacker is able to obtain one of these session keys, it will not be able to decrypt messages exchanged in other sessions.

[3]Hereafter, NO and STTP will be used as acronyms

- ***User anonymity :*** User anonymity is provided by adopting anonymous signature generation and verification protocols based on the direct anonymous attestation (DAA) scheme proposed by Chen et al. [16]. The DAA proposal is especially suitable for usage in hybrid mesh networks where efficient anonymous signature algorithms are required along with the user-controlled linkability option. Underlying scheme together with the developed join protocol allows a user to obtain a secret signing key where no single party, neither powerful network operator nor a trusted third party, other than the user herself is able to acquire and use this key to generate anonymous signatures.

  Furthermore, neither signatures generated by a legitimate user can be linked nor their originator can be identified by any single party, but the coalition of the network operator (NO) and the so-called semi-trusted third party (STTP). Although the network operator is able to capture signatures throughout the network, it cannot link any two of these signatures since it does not have secret signing keys of the network users or any valuable information it can use for this purpose. Besides, semi-trusted third party, which is required to provide users with a certificate/credential on their secret signing keys, therefore able to record credential-user identity pairs, also cannot link any signatures since the credentials that are presented to the verifiers are randomized in a way that two randomizations of the same credential do not reveal any information that leads one to link the corresponding signatures. Thus, in each authentication session, network user must re-randomize its credential to prevent linking of its signatures.

- ***User Accountability :*** User accountability is obtained through the use of two different protocols, one of which is designed for the identification of the user and the other one is used for the revocation of the secret signing key, thus the user herself. These protocols are designed as two-party protocols to be performed by the NO and the STTP. Neither of these two authorities alone is able to perform these protocols in order to identify or revoke a user by itself. Consequently, if, for instance, the NO suspects malicious activity, she can report suspected user's signatures to the

STTP, which then initiates the identification protocol and thus starts an examination process for the corresponding user. Then if the user is found guilty of malicious activities, the STTP initiates the revocation protocol together with the NO. All communication between the NO and the STTP is authenticated and secured by conventional cryptographic means since privacy providing solutions are not needed between these two well-known parties.

The anonymous authentication and key agreement framework proposed in this work, which is called A^2-MAKE[4], provides legitimate users with network connection and/or services from anywhere without being identified or tracked[5]. Only the two semi-trusted entities, the NO together with the STTP can identify the creator of a given signature and/or determine whether or not any two of the given signatures are generated by the same signer.

1.3.1 Contributions

Contributions of this thesis can be summarized as follows;

i. Our framework provides both accountability and strong anonymity for users in wireless mesh networks.

ii. The protocols in our framework are shown to be efficient in terms of communication and computational complexities.

iii. Our three-party *Join* protocol helps reconcile the user privacy in the strongest sense and user accountability in an efficient and scalable manner in the same framework.

iv. The two-party identification protocol can be used to identify users without revealing their private keys whenever deemed necessary.

v. The two-party key revocation protocol can be used to revoke users in a controlled manner and prevents abuse by a single authority.

[4] abbreviation for Anonymous and Accountable Mutual Authentication and Key agrEement

[5] With user consent, A^2-MAKE framework allows the user to be tracked.

vi. Security assumptions on the trusted third party and the network operator are relaxed compared to previous solutions, making ours easier to deploy in realistic settings.

vii. The user accountability feature proposed in this thesis is implemented through user identification and revocation protocols. This feature assists catching misbehaving users trying to abuse anonymity infrastructure and is especially useful protecting against malicious activities such as Sybil attacks [15].

viii. Optional user-controllable linkability, which temporarily removes unlinkability requirement, is used to trace users for a time period. This option is useful for user convenience, but can be a necessity in certain situations. It can also be utilized in preventing anonymity based attacks.

ix. Anonymous authentication protocol is more efficient than similar protocols in literature in terms of computational complexity which dominates its execution time. For higher security levels it is expected to become more efficient.

x. Implementation and simulation results of the anonymous authentication protocol are provided in detail demonstrating the suitability of our proposed framework in practical settings.

Following are the *publications* which benefitted from the content of this thesis;

- A.O. Durahim, and E. Savaş. A-make: An efficient, anonymous and accountable authentication framework for wmns. In *Internet Monitoring and Protection (ICIMP), 2010 Fifth International Conference on*, pages 54-59, may 2010.
- A.O. Durahim, and E. Savaş. A^2-make: An efficient anonymous and accountable mutual authentication and key agreement protocol for wmns. *Ad Hoc Networks*, 9(7):1202-1220, 2011.

1.4 Summary of the Thesis

In the current chapter we summarize prior work, provide the main motivation and contributions of this thesis along with fundamental background information about related topics

such as wireless mesh networks.

In Chapter 2, mathematical preliminaries are given. First, notations used throughout the thesis are introduced and then number-theoretic hard problems and corresponding assumptions are provided. Finally, signature proofs of knowledge protocols are given and some are illustrated using examples. Furthermore, we discuss how the proof of knowledge protocols are employed as basic protocols in group signature and related schemes.

In Chapter 3, we introduce elliptic curve cryptography and pairing based cryptography that are being extensively used in our protocols. We mention elliptic curves defined over finite field and type of attacks on elliptic curve cryptosystems. Then, we introduce the bilinear pairings and available pairing implementations proposed to obtain efficient pairing based cryptosystems. In the end, we discuss pairing-friendly elliptic curves and related constructions.

In Chapter 4, we elaborate on the concept of group signatures, together with a related scheme called direct anonymous attestation. In this chapter, we provide historical background about group signatures and direct anonymous attestation schemes along with a discussion on the groundbreaking proposals for them. We first explore properties and security requirements of group signature schemes and then provide the preliminary constructions. Furthermore, we describe the improvements made possible by either reducing signature sizes, increasing the efficiency of protocols, or providing additional security features relevant in certain applications. We also discuss revocation mechanisms proposed for group signatures and then illustrate pairing-based group signature schemes. In the final section, we summarize direct anonymous attestation proposals as a popular variant of group signatures.

Chapter 5 comprises the main contribution of this thesis. In this chapter, we first discuss the main motivation for the development of an anonymous and accountable authentication and key agreement scheme named A^2-MAKE, and then give construction details of the proposed scheme designed specifically for hybrid wireless mesh networks. Then, we review the security and performance of this scheme and compare our approach with related work on this subject. Finally, we describe implementation and simulation details of the proposed protocols and provide the results of our timing analyses.

In Conclusion section, we summarize the results and achievements of this thesis along with directions for future research.

Chapter 2

Foundations and Basic Protocols

In this section, we provide notations used throughout this thesis, review cryptographic hard problems and introduce the concept of signature proof of knowledge.

2.1 Notations and Preliminaries

Throughout this thesis, integers, group elements, and strings are all assumed to be represented in binary form. The symbol $||$ denotes the concatenation of two strings or string representation of integers or group elements. For A being a set, $a \in_R A$ means that a is chosen randomly from the set A, and a is assumed to be distributed uniformly. For an integer n, $\mathbb{Z}_n$ denotes the ring of integers modulo n and $\mathbb{Z}_n^*$ denotes the multiplicative group modulo n which is comprised of invertible elements. For a cyclic group G of order n, $G = \langle g \rangle$ means that g is the generator of group G, with order n. The number of elements in this group, G, is denoted by $|G|$, where $n = |G|$.

$\mathbb{F}_q$ denotes a finite field of order q and $\mathbb{F}_q^*$ denotes the multiplicative group of nonzero elements of $\mathbb{F}_q$, which can be stated equivalently as $\mathbb{F}_q^* \equiv \mathbb{F}_q \backslash \{0\}$. Similarly $\overline{\mathbb{F}}_q$ denotes the algebraic closure of finite field $\mathbb{F}_q$.

$\mathcal{H}(\cdot)$ denotes a hash function that maps binary representation of elements of a group, strings and/or integers to fixed-length binary strings. For example, $\mathcal{H} : G \rightarrow \{0,1\}^k$ means that hash function takes binary representation of group elements from G as input and maps it into binary string of length k.

We denote by $c[i]$, the *i-th* bit of the binary string c, where one starts counting from

the right-hand end. For example, if $c = 10011$, then $c[2] = 1$ and $c[3] = 0$.

If not stated otherwise, $\log(x)$, denotes the logarithm of x with respect to base 2 and $\lceil \log(x) \rceil$ is the bit-length of the number x.

QR(n) denotes quadratic residue modulo n[1]. An RSA modulus $n = pq$ is safe if its prime factors are of the form, $q = 2q' + 1$ and $p = 2p' + 1$ where p' and q' are also prime numbers.

2.2 Number Theoretic Assumptions

In the following, number theoretic problems and corresponding assumptions are given. They are both applicable to cyclic subgroups of a multiplicative group of a finite field and elliptic curve group defined over a finite field, etc. Let G be a finite cyclic group of order q $(= |G|)$, and g be its generator, $G = \langle g \rangle$.

Definition 1 ***Discrete Logarithm Problem (DLP) :*** *Given elements g and y, find an integer $k \in \mathbb{Z}_q^*$ such that $y = g^k$, if such an integer exists. k is called the* ***discrete logarithm*** *or* ***index*** *of element y with respect to g, denoted by $\log_g(y)$ $(= ind_g y)$.*

Using the same terminology, computational and decision Diffie-Hellman (CDH and DDH, respectively) problems in the same group can be defined as follows;

Definition 2 ***Diffie-Hellman Problem (DHP-CDHP) :*** *Given elements g, g^a, g^b where $a, b \in \mathbb{Z}_q^*$, compute g^{ab}.*

Definition 3 ***Decision Diffie-Hellman Problem (DDHP) :*** *Given elements g, $h = g^a$, $y = g^b$, $z = g^c$ where $a, b, c \in \mathbb{Z}_q^*$, decide if $g^c = g^{ab}$ (or equally decide if $z = y^a$).*

Corresponding *Decisional Diffie-Hellman assumption* was first explicitly mentioned in [17] and one can refer to [18] for an in-depth discussion. CDH and DDH assumptions state that it is computationally infeasible to solve their corresponding problems. Note that DDHP is easier than the (C)DHP which involves finding g^{uv} from g^u and g^v. Thus, DDH

[1]Note that deciding whether some y is in QR(n) is believed to be infeasible if the factorization of n is unknown.

assumption is a stronger assumption. Both DDH and CDH assumptions are stronger than the assumption that computing discrete logarithm is hard. That is to say, if one is able to solve DLP, one can also solve both CDHP and DDHP: given $y = g^a,\ z = g^b,\ t = g^c$, first solve DLP for y and z and then use corresponding integers a and b to compute g^{ab}, and then check if $g^{ab} = t$.

Other related hard problems are defined similarly as follows;

Definition 4 ***Double Discrete Logarithm Problem (DDLP) :*** *Given elements $g,\ y \in G$, and $a \in \mathbb{Z}_q^*$, find an integer $k \in \mathbb{Z}_q^*$ such that $y = g^{(a^k)}$, if such an integer exists. k is called as the* ***double discrete logarithm*** *of element y with respect to bases a and g, denoted by* $\log_a(\log_g y)$.

Definition 5 ***eth-Root Discrete Logarithm Problem :*** *Given elements $g,\ y \in G$, find an integer $k \in \mathbb{Z}_q^*$ such that $y = g^{(k^e)}$, if such an integer exists. k is called as the e**th-root of discrete logarithm*** *of element y with respect to g.*

Double discrete logarithms and eth-root of discrete logarithms are first defined and used in group signature schemes proposed by Stadler [19] and Camenisch and Stadler [1], respectively.

Definition 6 ***Representation Problem (RP) :*** *Given elements $g_1, g_2, \ldots, g_k, h \in G$, compute integers $a_1,\ a_2, \ldots,\ a_k \in \mathbb{Z}_q^*$, such that $h = g_1^{a_1} g_2^{a_2} \cdots g_k^{a_k}$. Problem is defined in [17].*

Definition 7 ***LRSW Problem :*** *Given elements $g,\ X = g^x,\ Y = g^y$ where $x, y \in \mathbb{Z}_q^*$, compute triple (a, a^y, a^{x+xys}) for a given integer $s \neq 1,\ s \in \mathbb{Z}_q^*$ where $a \in_R G$ is a random element, $a = g^k$ and $k \in_R \mathbb{Z}_q^*$. Here, one is also given access to an Oracle which returns such a triple for any queried integer z that is different from the s in question.*

LRSW assumption is introduced by Lysyanskaya et al. [20], which states that it is infeasible for a computationally bounded adversary to solve the corresponding LRSW problem.

Integer factorization is another number-theoretical problem where it is computationally hard to factor a given large composite number to its prime factors, $N = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$.

In the following, we state RSA and related problems which utilize this well known problem.

Definition 8 ***RSA Problem :*** *Given a large composite number N = pq, where* $p,\ q$ *are large primes, an exponent* e *where* $2 < e < N$*, and ciphertext* $C \in Z_N^*$*, find* P *such that* $C = P^e \pmod N$*. This problem is based on the hardness of computing* e*th-root when the integer factorization of the modulus is unknown and the hardness of factoring the modulus itself.*

RSA cryptosystem is invented by Rivest et al. [6], which is based on the *RSA assumption*[2] which states that it is computationally infeasible to solve the RSA problem when the modulus is generated randomly and sufficiently large and message P is also random. Following is the related strong RSA problem which can be solved if one finds an algorithm that solves the original RSA problem.

Definition 9 ***Strong RSA Problem :*** *Given a random and sufficiently large RSA modulus* n *and* $c \in Z_n^*$*, find a pair* $(u, e) \in Z_n^* \times Z$ *such that* $u^e = c$ *and* $e > 1$*.*

The *Strong RSA assumption* states that it is computationally infeasible, on given a random RSA modulus n and $c \in Z_n^*$, to find pair $(u, e) \in Z_n^* \times Z$. Strong-RSA assumption was introduced by Baric and Pfitzmann [22] and Fujisaki and Okamoto [23] and later on various signature schemes (cf. [24]) are based on this number-theoretic assumption.

Definition 10 ***Modified Strong RSA Problem :*** *Given* $G,\ z \in G$ *and* $M \subset M(G, z)$ *with* $|M| = O(l_g)$*, find a pair* $(u, e) \in G \times Z$ *such that* $u^e = z,\ e \in \left\{2^{l_1} - 2^{\tilde{l}}, ..., 2^{l_1} - 2^{\tilde{l}}\right\}$ *and* $(u, e) \notin M$ *where* $\tilde{l} = \epsilon(l_2 + k) + 1$ *and* $\epsilon > 1$ *and* $k, l_1, l_2 < l_g$ *and* $M(G, z) = \left\{(u, e) | z = u^e, u \in G, e \in \left\{2^{l_1} - 2^{l_2}, ..., 2^{l_1} - 2^{l_2}\right\}, e \in primes\right\}$.

Although the assumption that breaking modified strong RSA problem is infeasible was introduced in [25, 26], a similar assumption was also proposed in [22], such that e is required to be a prime but the size of the exponents has no restriction.

Modified strong RSA problem is at least as hard as strong RSA problem due to the range restriction on the exponents.

[2] see Rivest and Kaliski [21] for an in-depth discussion

2.3 Signature Proof of Knowledge

Signature proof of knowledge is used as building blocks in anonymous authentication and privacy preserving signature schemes, e.g. group signature, direct anonymous attestation. Actually, these proofs are all related to proving the knowledge of a secret which is cryptographically protected based on the hardness of some number theoretic problem.

In this work, we will follow the notation introduced by Camenisch and Stadler [1] for various proof of knowledge of discrete logarithms and of the validity of statements about discrete logarithms. To give an example;

$$PK\left[(\alpha,\beta) : y_1 = g^{\alpha} \ \wedge \ y_2 = g^{\beta}h^{\alpha} \ \wedge \ \alpha \in [a,b]\right]$$

denotes a zero-knowledge proof of knowledge of integers α and β such that $y_1 = g^{\alpha}$ and $y_2 = g^{\beta} \cdot h^{\alpha}$ holds where $a \leq \alpha \leq b$, and g and h are generators of a group G. The convention used here is that Greek letters represent values that are being proven to be known, while remaining values are the ones that are already known by the verifier.

These are the honest-verifier zero-knowledge proofs of knowledge which can be turned into signatures by applying techniques known as Fiat-Shamir heuristic [27, 28]. There, the verifier is replaced by a suitable hash function and the challenge is obtained using the commitment value as one of the arguments to this hash function. This construction leads to a security model formalized as random oracle methodology, [29, 30, 31]. Following is the notation used for signature proof of knowledge[3] on a message m, corresponding to the proof of knowledge given above;

$$SPK\left[(\alpha,\beta) : y_1 = g^{\alpha} \ \wedge \ y_2 = g^{\beta}h^{\alpha} \ \wedge \ \alpha \in [a,b]\right](m)$$

In nearly all but the initial proposals of the group signature schemes, SPKs are utilized for proving the knowledge of a secret on which a membership certificate is granted by a designated group authority. This SPK along with the corresponding certificate proves the membership of a user to that respective group. In the following, we provide implementa-

[3] Abbreviated as SPK from now onward

tion details of various SPKs mentioned throughout this work;

1. **SPK of Discrete Logarithm**

$$\text{A pair } (c, s) \in \{0,1\}^k \times \mathbb{Z}_q \quad \text{satisfying} \quad c = \mathcal{H}(m||g||y||g^s y^c)$$

is a *signature proof of knowledge of discrete logarithm* of element $y \in G$ to the base g on a message m[4]. Such a signature is denoted by

$$\text{SPKDL}\left[(\alpha) : y = g^{\alpha}\right](m)$$

and can be computed if the secret value x, which is the discrete logarithm of y to the base g, is known as follows:

Select $r \in_R \mathbb{Z}_q$ randomly and compute $t = g^r$, then use these values to compute the challenge and corresponding response as;

$$c = \mathcal{H}(m||g||y||t) \text{ and } s = r - cx \pmod{q}$$

The verifier of such a signature (c, s) with respect to public key y of the signer should;

compute $t' = g^s y^c$ and then check if $c = \mathcal{H}(m||g||y||t')$.

SPKDL is introduced by Schnorr [32], Chaum et al. [33] and shown to be zero-knowledge by Damgård [34]. Here, the protocol between prover and verifier is a honest-verifier non-interactive zero knowledge protocol where g^r, c, and s are commitment, the challenge and the response values, respectively, which are all generated by the prover, and they are analogues to the values used in interactive zero knowledge protocols, where the challenge c is supplied to the prover by the verifier.

[4]This is actually the Schnorr signature [32] where input to the hash function is slightly different

2. **SPK of the Equality of Two Discrete Logarithms**

$$\text{A pair } (c, s) \in \{0,1\}^k \times \mathbb{Z}_q \quad \text{satisfying} \quad c = \mathcal{H}(m||g||y||h||z||g^s y^c||h^s z^c)$$

is a *signature proof of knowledge of the equality of two discrete logarithms* of group elements $y, z \in G$ with respect to the bases $g, h \in G$, respectively on a message m. Such a signature is denoted by

$$\text{SPKEQDL}\left[(\alpha) : y = g^\alpha \ \wedge \ z = h^\alpha\right](m)$$

and can be computed as follows, if the secret value x, which is the discrete logarithm of y and z to the bases g and h, respectively, is known:

Select $r \in_R \mathbb{Z}_q$ randomly and compute values c and s as;

$$c = \mathcal{H}(m||g||y||h||z||g^r||h^r) \ \text{ and } \ s = r - cx \pmod q$$

SPKEQDL is introduced and used first in Chaum [35], Chaum and Pedersen [36]. This signature can be seen as a two parallel signature knowledge of discrete logarithms,

$$SPKDL\left[(\alpha) : y = g^\alpha\right](m) \quad \text{and} \quad SPKDL\left[(\alpha) : z = h^\alpha\right](m),$$

where the exponent for the commitment, and the challenge and response values are the same.

3. **SPK of One out of Two Discrete Logarithms**

A 4-tuple $(c_1, c_2, s_1, s_2) \in \{0,1\}^k \times \{0,1\}^k \times \mathbb{Z}_q^2$ satisfying

$$c_1 \oplus c_2 = \mathcal{H}(m||g||h||y_1||y_2||g^{s_1} y_1^{c_1}||h^{s_2} y_2^{c_2})$$

is a *signature of knowledge of the discrete logarithm of (at least) one group element*

out of two (y_1, y_2) to the bases (g, h), respectively on a message m. Such a signature is denoted by

$$\text{SPKONEOUTTWO}\left[(\alpha_1, \alpha_2) : y_1 = g^{\alpha_1} \ \wedge \ y_2 = h^{\alpha_2}\right](m)$$

and can be computed as follows;

Using secret key x_1, select randomly $r_1, \ s_2 = r_2, \ c_2 \in_R \mathbb{Z}_q$ and compute $t_1 = g^{r_1}$ and $t_2 = h^{r_2} y_2^{c_2}$ and then using these values compute c_1 and s_1 as,

$$c_1 = c_2 \oplus \mathcal{H}(m||g||h||y_1||y_2||t_1||t_2)$$

$$s_1 = r_1 - x_1 c_1 \pmod{q}$$

SPKONEOUTTWO is introduced by Cramer et al. [37] and also utilized in group signature scheme proposed by Camenisch and Michels [26].

4. **SPK of One out of Many Discrete Logarithms**

The previous SPK can be generalized to proving the knowledge of one out of many discrete logarithms (cf. [38]) as follows;

A $2n$ tuple $(c_1, ..., c_n, s_1, ..., s_n) \in (\{0,1\}^k)^n \times \mathbb{Z}_q^n$ satisfying

$$\bigoplus_{i=1}^{n} c_i = \mathcal{H}(m||g||y_1||...||y_n||g^{s_1} y_1^{c_1}||...||g^{s_n} y_n^{c_n})$$

is a *signature of knowledge of the discrete logarithm of (at least) one group element out of many* $\{y_1, ..., y_n\}$ to the base g on a message m. Such a signature is denoted by

$$\text{SPKONEOUTMANY}\left[(\alpha_i)_{i=1,...,n} : \bigwedge_{i=1,...,n} y_i = g^{\alpha_i}\right](m)$$

and can be computed as follows;

Using secret key x_1, select randomly $r, s_2, \ldots, s_n, c_2, \ldots, c_n \in_R \mathbb{Z}_q$ and compute $t_1 = g^r$ and $t_i = g^{s_i} y_i^{c_i}$ for $i = 2, \ldots, n$ and then using these values compute c_1 and s_1 as,

$$c_1 = \bigoplus_{2}^{n} c_i \oplus \mathcal{H}(m||g||y_1||...||y_n||t_1||...||t_n)$$

$$s_1 = r - x_1 c_1 \pmod{q}$$

5. **SPK of Representation**

A (n+1) tuple $(c,\ s_1,\ \ldots,\ s_n) \in \{0,1\}^k \times \mathbb{Z}_q^n$ satisfying

$$c = \mathcal{H}(m||g||\ldots||g_n||y||y^c \prod_{i=1}^{n} g_i^{s_i})$$

is a *signature of knowledge of representation* (cf. [33]) of y to the bases $g_1, \ldots, g_n$ on a message m. Such a signature is denoted by

$$\text{SPKREP}\left[(\alpha_i)_{i=1,\ldots,w} : y = \prod_{i=1}^{n} g^{\alpha_i}\right](m)$$

and can be computed as follows;

Choose $r_i \in_R \mathbb{Z}_q$ randomly for $i = 1, \ldots, n$ and compute $t = \prod_{i=1}^{n} g_i^{r_i}$, and then using these values compute c and s_i values as,

$$c = \mathcal{H}(m||g_1||...||g_n||y||t)$$

$$s_i = r_i - x_i c \pmod{q},\ i = 1,\ \ldots,\ n.$$

SPKREP is introduced by Brands [17] along with its corresponding representation problem (cf. Section 2.2 - 6).

6. **SPK of Double Discrete Logarithms**

Let $n \leq k$ be a security parameter. An $(n+1)$ tuple (c, s_1, ..., s_n) $\in \{0,1\}^k \times \mathbb{Z}_q^{*n}$

satisfying the equation

$$c = \mathcal{H}(m||y||g||a||t_1||...||t_n) \text{ with } t_i = \begin{cases} g^{(a^{s_i})} & \text{if c[i] = 0} \\ y^{(a^{s_i})} & \text{otherwise} \end{cases}$$

is a *signature proof of the knowledge of a double discrete logarithm* of y to the bases g and a, on a message m. Such a signature is denoted by

$$\text{SPKLOGLOG}\left[\alpha : y = g^{(a^{\alpha})}\right](m)$$

Computation can be started by choosing an x with an upper bound on its length ($0 \leq x < 2^q$). Choosing $r_i \in_R \{0, ..., 2^q - 1\}$ and computing $t'_i = g^{(a^{r_i})}$ for $i = 1, \ldots, n$, one computes values c and s_i, where $i = 1, \ldots, n$ as,

$$c = \mathcal{H}(m||y||g||a||t'_i||...||t'_n)$$

and

$$s_i = \begin{cases} r_i \pmod q & \text{if c[i] = 0} \\ r_i - x \pmod q & \text{otherwise.} \end{cases}$$

SPKLOGLOG is utilized in various protocols [1, 19, 39, 40].

7. **SPK of *e-th* Root of Discrete Logarithm**

The last building block in this section is the signature based on the proof of knowledge of *e-th* root of a discrete logarithm. This SPK is utilized in Camenisch and Stadler [1] to generate a signature on a secret which is the *e-th* root of a discrete logarithm of a given publicly known number.

Let $n \leq k$ be a security parameter. An $(n+1)$ tuple $(c, s_1, \ldots, s_n) \in \{0,1\}^k \times \mathbb{Z}_q^{*n}$

satisfying the equation

$$c = \mathcal{H}(m||y||g||e||t_1||...||t_n) \text{ with } t_i = \begin{cases} g^{(s_i{}^e)} & \text{if c[i] = 0} \\ y^{(s_i{}^e)} & \text{otherwise} \end{cases}$$

is a *signature proof of the knowledge of an e-th root of discrete logarithm* of y with respect to the base g, on a message m. Such a signature is denoted by

$$\text{SPKROOTLOG}\left[\alpha : y = g^{\alpha^e}\right](m)$$

This can be computed if the *e-th* root x of discrete logarithm of y to the base g is known. For randomly chosen $r_i \in \mathbb{Z}_q^*$ for $i = 1, \ldots, n$, one computes the values t_i' = $g^{(r_i{}^e)}$, and then computes c and s_i values as,

$$c = \mathcal{H}(m||y||g||e||t_i'||...||t_n')$$

$$s_i = \begin{cases} r_i & \text{if c[i] = 0} \\ r_i/x \pmod q & \text{otherwise.} \end{cases}$$

Chapter 3

Elliptic Curve and Pairing Based Cryptography

In this chapter, we introduce and give necessary information on elliptic curve and pairing-based cryptosystems.

3.1 Elliptic Curve Cryptography

Elliptic curve cryptography is introduced by Koblitz [41] and Miller [42], where they propose constructing public key cryptosystems based on group of points on an elliptic curve defined over a finite field. As a result, elliptic curves defined over finite fields are used to build public key cryptosystems that allow making use of small sized keys whereby more efficient cryptographic schemes can be proposed than the ones utilizing multiplicative groups over finite fields.

3.1.1 Elliptic Curves over Finite Fields

An elliptic curve E defined over a finite field $\mathbb{F}_q$ is denoted by $E(\mathbb{F}_q)$[1] where q is a prime power, $q = p^m$, and p is the prime characteristic of the underlying finite field. An elliptic curve group can be defined by the points (x, y) where $x, y \in \mathbb{F}_q$ satisfying the Generalized Weierstrass equation

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{3.1}$$

[1] For further information on elliptic curves and their usage in cryptography, one can refer to Silverman [43], Blake et al. [44]

where $a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}_q$ together with an additional point $\mathcal{O}$, called *point at infinity*, which serves as the identity element of the group.

The Weierstrass equation can be transformed into simpler forms by linear change of variables according to the characteristic p of the base field $\mathbb{F}_q$. For example, taking the prime characteristic of the underlying field, $p > 3$, equation simplifies to

$$y^2 = x^3 + ax + b \tag{3.2}$$

where $a, b \in \mathbb{F}_q$ and $4a^3 + 27b^2 \neq 0$. Here, the last requirement, the discriminant[2] having a value other than zero, is necessary to avoid singular elliptic curves and obtain non-singular ones, i.e. having distinct roots. If the discriminant is equal to zero, then the resulting elliptic curve is singular which makes elliptic curve addition being either addition of elements in $\mathbb{F}_q$ or multiplication of elements in $\mathbb{F}_q^*$ or in a quadratic extension of $\mathbb{F}_q$. Consequently, powerful algorithms designed to solve discrete logarithm problem in finite fields also become applicable to elliptic curve groups[3].

Elliptic curve points satisfying the above equation together with the point at infinity, $\mathcal{O}$, form an abelian group under the elliptic curve point addition as group arithmetic defined by so-called "chord-tangent rule"[4]. The number of points on this group, $E(\mathbb{F}_q)$, also called the cardinality of the group, is denoted by $\#E(\mathbb{F}_q)$.

An important theorem by Hasse on the number of points on an elliptic curve is given in the following;

Theorem 11 (Hasse's theorem)
Let $E(\mathbb{F}_q)$ be an elliptic curve defined over finite field $\mathbb{F}_q$. Then, the cardinality (order) of $E(\mathbb{F}_q)$, #E is defined as $\#E(\mathbb{F}_q) = q + 1 - t$, where $|t| \leq 2\sqrt{q}$.

Here t is called the *trace of Frobenious*. From this theorem, we can deduce that the cardinality of the elliptic curve is close to the size of the underlying field.

Following is the theorem by Weil which makes it easier to find the number of points

[2] Actually the discriminant is given by $\Delta = -16(4a^3 + 27b^2)$
[3] For an in-depth discussion on the subject, refer to Section 2.10 of [45]
[4] cf. Chapter III of [44]

on an elliptic curve defined over an extension field, $\mathbb{F}_{q^k}$;

Theorem 12 (Weil's theorem)
Let t = q + 1 - #$E(\mathbb{F}_q)$ where $q = p^m$ and p is prime. Then,

$$\#E(\mathbb{F}_{q^k}) = q^k + 1 - (\alpha^k + \beta^k) \tag{3.3}$$

where α, β can be found by factoring the polynomial $x^2 - tx - q$ as $(x - \alpha)(x - \beta)$ over the field of complex numbers. This can be restated recursively as;

$$t_n = t_1 t_{n-1} - q t_{n-2} \tag{3.4}$$

where $t_0 = 2$ and $t_1 = q + 1 - \#E(\mathbb{F}_q)$ and the number of points on the curve is #$E(\mathbb{F}_{q^k}) = q^k + 1 - t_k$

Following two theorems are related to characterization of the elliptic curve groups. First one is due to Waterhouse [46];

Theorem 13 *Let $q = p^m$ be a prime power and let #$E(\mathbb{F}_q) = q + 1 - t$. Then, there exists an elliptic curve $E(\mathbb{F}_q)$ defined over finite field $\mathbb{F}_q$ if and only if $|t| \leq 2\sqrt{q}$ and t satisfies one of the following;*

1. $t \not\equiv 0 \pmod{p}$ *and* $t^2 \leq 4q$
2. m *is odd and one of the following holds;*
   (a) $t = 0$
   (b) $t^2 = 2q$ *and* $p = 2$
   (c) $t^2 = 3q$ *and* $p = 3$
3. m *is even and one of the following holds;*
   (a) $t^2 = 4q$
   (b) $t^2 = q$ *and* $p \not\equiv 1 \pmod{3}$

(c) $t = 0$ *and* $p \not\equiv 1 \pmod 4$

Here, the first condition pertains to ordinary elliptic curves whereas the other two conditions are related to the supersingular curves (cf. Section 3.2.4).

Second theorem is due to Ruck [47] which describes group structure of the elliptic curves;

Theorem 14 *Let* $\#E(\mathbb{F}_q)$ *be the order of an elliptic curve E defined over* $\mathbb{F}_q$ *and let* $\#E(\mathbb{F}_q) = p^e n_1 n_2$ *with* $p \not| \, n_1 n_2$ *and* $n_1 | n_2$*. Then, there exists* E *over* $\mathbb{F}_q$ *such that*

$$E(\mathbb{F}_q) \cong \mathbb{Z}_{p^e} \oplus \mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2}$$

if and only if

- $n_1 | \, q - 1$ *in cases of items given in Theorem 13 except 3a*
- $n_1 = n_2$ *in case of 3a of Theorem 13*

3.1.2 Elliptic Curve Cryptosystems

Elliptic curve cryptosystems are built on the cryptographic hardness assumptions that are analogous to the finite field counterparts. The first one is the elliptic curve discrete logarithm assumption which states that it is computationally infeasible to solve the corresponding problem defined as follows:

Definition 15 ***ECDLP :*** *Let* G *be an elliptic curve group of order* q *and let point* P *be its generator,* $G = \langle P \rangle$*. Given points* P *and* Q*, find an integer* $k \in \mathbb{Z}_q^*$ *such that* $Q = kP$*, if such an integer exists.* k *is called as the discrete logarithm of point* Q *with respect to point* P*.*

Using the same terminology, computational and decision Diffie-Hellman problems in elliptic curve groups can be defined as follows;

Definition 16 ***ECDHP :*** *Let* G *be an elliptic curve group of order* q *and let point* P *be its generator,* $G = \langle P \rangle$*. Given points* $P, \ aP, \ bP$ *where* $a, b \in \mathbb{Z}_q^*$*, compute* abP*.*

Definition 17 ***ECDDHP :*** *Let G be an elliptic curve group of order q and let point P be its generator, $G = \langle P \rangle$. Given points $P,\ Q = aP,\ R = bP,\ S = cP$ where $a, b, c \in \mathbb{Z}_q^*$, decide if $cP = abP$ (or equally decide if $S = aR$).*

3.1.3 Attacks on Elliptic Curves

Beginning with the introduction of elliptic curve cryptography, attacks have been devised to solve the discrete logarithm and other related problems. Most of them are the adaptation of the attacks discovered for solving analogous problems on multiplicative groups defined over finite fields to elliptic curve groups. Well-known attacks can be summarized as follows;

- Generic Attacks

  – *Pohlig-Hellman Attack (PHA) :* Pohlig-Hellman algorithm [48] reduces the discrete logarithm problem ($k = log_P Q$) in elliptic curve group of order q to computing this problem in prime order subgroups. PHA works as follows;
  Let prime factorization of the order of the curve is $q = p_1^{e_1} p_2^{e_2} \cdots p_t^{e_t}$. Then PHA strategy is to compute $k_i = k \pmod{p_i^{e_i}}$ for each $i \in [1, t]$ and then to solve the resulting system of congruences using Chinese remainder theorem that gives a unique solution.

  Therefore, in order to resist PHA, one needs to select a curve with order divisible by a large prime, perhaps a prime order curve.

  – *Pollard's rho Attack :* Pollard's rho algorithm, with the purpose of finding a solution to the discrete logarithm problem, ($k = log_P Q$), tries to find distinct pairs of integers (a, b) and (c, d), where $a, b, c, d \in \mathbb{Z}_q$, such that $aP + bQ = cP + dQ$. If such pairs exist, then one can continue by transforming the equation into

$$aP - cP = dQ - bQ \longrightarrow (a - c)P = (d - b)Q$$

which then results in

$$(a-c)P = (d-b)kP \;\longrightarrow\; (a-c) \equiv (d-b)k \pmod{q}$$

Consequently, k can be computed as,

$$k = (a-c)(d-b)^{-1} \pmod{q}.$$

In order to find such pairs one may naively select integer pairs (a', b') and compute $R = a'P + b'Q$ and store the triple $(a',\ b',\ R)$ until one finds the same point R with different pair of integers, known as the collision. Expected number of tries for finding such a collision is given by $\sqrt{\pi q/2} \approx 1.25\sqrt{q}$. In Pollard's rho method, storage problem of these triples computed for the naive approach is overcome by using a suitable iterating function (cf. Section 4.1.2 of Hankerson et al. [49]).

Pollard's rho attack can be parallelized and, with the help of automorphisms, the expected running time can be reduced to $\approx \frac{1}{2S}\sqrt{q}$ where S is the number of processing units. As a result, parallelization reduces time linearly and due to Pollard's rho attack, ECDLP problem can be solved in subexponential, $O(\sqrt{n})$, time.

- Specialized Attacks

  - *Anomalous Curve Attack :* An elliptic curve $E(\mathbb{F}_q)$ is said to be anomalous if it has prime order q, that is $\#E(\mathbb{F}_q) = q$. As a result, $E(\mathbb{F}_q)$ is a cyclic group of order q and isomorphic to the additive group of integers, $\mathbb{F}_q^+$, modulo q. ECDLP is then reduced to finding $k \in [0, q-1]$ such that $b \equiv ka \pmod{q}$ where $a, b \in \mathbb{F}_q^+$ which can be solved efficiently using extended Euclidean algorithm (cf. Algorithm 2.19 in Hankerson et al. [49]). So, together with an efficient automorphism $\psi : E(\mathbb{F}_q) \to \mathbb{F}_q^+$ which is shown independently by Smart in [50], ECDLP can be solved in polynomial-time. Therefore, one must avoid using anomalous curves in cryptographic applications.

– *Pairing Attacks :* The logic behind pairing attacks is to use pairings in a way to reduce discrete logarithm problem in an elliptic curve group to corresponding problem in multiplicative group of an extension field of the underlying finite field. By this way, one is able to utilize powerful algorithms discovered for solving the finite field discrete logarithm problem which cannot be applicable to elliptic curve groups. Menezes et al. [51] and Frey and Rück [52] came up with the idea of using Weil and Tate pairings for this purpose, respectively. So, one first chooses a suitable bilinear pairing (cf. Section 3.2.1), e, such that $s = e(P, Q)$, $s \in \mathbb{F}^*_{q^k}$, $P, Q \in E(\mathbb{F}_q)$ and $e(P, Q) = e(P, tP) = e(P, P)^t = g^t$. Then, in order to find $t = log_P Q$, one solves the discrete logarithm of s with respect to base g, which is the generator of a cyclic subgroup of the extension field, $\mathbb{F}_{q^k}$. These attacks led to a new field of cryptography, called Pairing-based Cryptography (cf. Section 3.2).

As mentioned above, the best method for solving elliptic curve discrete logarithm problem requires $O(\sqrt{n})$ time, where n is the order the group. So, in order to obtain 80-bit security level [53], one requires group order of approximately 160-bits in length to resist these attacks. In contrast, 1024-bit order groups are required in finite fields in order to obtain the same level of security, due to specialized subexponential algorithms.

3.2 Pairing Based Cryptography

Pairing based cryptography uses primitives known as (bilinear) pairings in designing and constructing cryptographic algorithms and protocols. A pairing is a function which maps a pair of points from an elliptic curve to an element of a multiplicative subgroup of a finite field.

Pairings are initially used in attacking elliptic curves. The idea is to reduce the elliptic curve discrete logarithm problem into discrete logarithm problem in finite field, using modified Weil pairing (known as MOV attack [51]) or Tate pairing (known as FR attack [52]).

Then, pairings are proposed for constructive use for the first time by Joux [54] in one-round three-party Diffie-Hellman key exchange protocol. After that, it is used to build the first practical identity based encryption scheme [55]; a breakthrough in the field that solves a nearly two decade old open problem (cf. [56]) in an efficient manner.

After these proposals, pairing operation has emerged as an important cryptographic primitive and many recent protocols utilize it. Examples include non-interactive key agreement schemes [57], group signature schemes [58, 59, 60], [61], traitor tracing schemes [62], identity-based ring signature schemes [63], and last but not the least direct anonymous attestation schemes [16, 64, 65, 66, 67, 68, 69].

3.2.1 Bilinear Pairings

A bilinear pairing operation can be defined as follows;

Let G_1, G_2 be two cyclic groups[5] of some large prime order q and P, Q be generators of these two groups, respectively. Furthermore, let G_M be multiplicative cyclic group (finite field group) of same prime order q. Then, $\hat{e} : G_1 \times G_2 \rightarrow G_M$ is a bilinear map, which satisfies the following properties;

1. *Bilinear :* $\forall P \in G_1$, and $\forall Q \in G_2$, and $\forall a, b \in \mathbb{Z}_q^*$, $\hat{\mathrm{e}}(aP, bQ) = \hat{\mathrm{e}}(P, Q)^{ab}$.[6]

2. *Non-degenerate :* There exist non-trivial elements $P \in G_1$ and $Q \in G_2$ such that $\hat{\mathrm{e}}(P, Q)$ is not the identity element of G_M, that is $\hat{\mathrm{e}}(P, Q) \neq 1$.[7]

3. *Computable :* There exists an efficient polynomial time algorithm to compute $\hat{\mathrm{e}}(P, Q)$ for all $P \in G_1$ and $Q \in G_2$.

There are two types of bilinear pairing settings, namely symmetric and asymmetric pairings. They are classified into three different categories according to [71] based on the relationship between the two input groups G_1 and G_2.

[5] Typically pairing friendly elliptic curve groups. In a more general pairing definition and usage, G_1 is assumed to be a cyclic group of prime order q, but G_2 is allowed to be non-cyclic group with the same prime order (cf. [70]).

[6] This can also be stated as, $\forall P, S \in G_1$, and $\forall Q, R \in G_2$, $\hat{\mathrm{e}}(P + S, Q) = \hat{\mathrm{e}}(P, Q)\ \hat{\mathrm{e}}(S, Q)$ and $\hat{\mathrm{e}}(P, Q + R) = \hat{\mathrm{e}}(P, Q)\ \hat{\mathrm{e}}(P, R)$.

[7] This can also be stated as, $\forall P \in G_1$, $\hat{\mathrm{e}}(P, Q) \neq 1$ if $Q \neq 1$, and $\forall Q \in G_2$, $\hat{\mathrm{e}}(P, Q) \neq 1$ if $P \neq 1$.

In the first type, the two input groups are the same, rendering the corresponding pairing symmetric. In the second type, they are different cyclic groups, but there exists an efficiently computable homomorphism from the second input group to the first one, $\psi : G_2 \rightarrow G_1$. If an efficiently computable homomorphism also exists from the first input group to the second one, then corresponding pairing is considered as symmetric, thus belongs to the first type. And the last type is the one where two input groups are different and there is no efficiently computable homomorphism between these two input groups. The last two types are considered as asymmetric pairings.

In summary,

- ***Type 1:*** $G_1 = G_2$, symmetric;
- ***Type 2:*** $G_1 \neq G_2$, but there is an efficiently computable homomorphism $\psi : G_2 \rightarrow G_1$, asymmetric;
- ***Type 3:*** $G_1 \neq G_2$, and there is no efficiently computable homomorphism between G_1 and G_2, asymmetric.

From here on, $\psi : G_2 \rightarrow G_1$ denotes a homomorphism from group G_2 to G_1 which becomes an isomorphism if one restricts both groups to be cyclic subgroups.

Symmetric pairings (Type 1) can be realized only by using appropriate supersingular elliptic curves. However, supersingular elliptic curves have embedding degrees up to 6 (cf. Section 3.2.4 - 3.2.4.1), which results in scalability problems. So, one must use ordinary elliptic curves as input to the pairing computations to attain higher embedding degrees. Because of the mapping available from G_2 to G_1 for Type 2 pairings, one can easily convert a scheme suggested under symmetric pairing setting to asymmetric one with minimal change in the security proofs. Here, security assumptions based on the input group G (=$G_1 \& G_2$) in symmetric pairing can be based on G_2 in the asymmetric counterpart. An important drawback of Type 2 pairing is the lack of a method to hash a string to an element of G_2 of which the discrete logarithm to a fixed base is unknown. Therefore, if hashing onto an element of group G_2 defined over an ordinary curve is required, then one must use Type 3 pairings. In Type 3 pairings, G_1 and G_M are cyclic groups of order q whereas G_2 is a group, where each element has order dividing q.

A crucial problem arises when one both needs to hash bit strings onto G_2 and to have efficiently computable homomorphism from group G_2 to G_1, i.e. the verifier-local revocation (VLR) group signature scheme proposed by Boneh and Shacham [58]. Then, one cannot use Type 2 pairing, the one utilized in the proposed scheme, since it does not allow one to hash bit strings securely onto group G_2. On the other hand, Type 3 pairings also cannot be employed since there does not exist an efficiently computable homomorphism from G_2 to G_1. Due to this fact, a new type of pairing is introduced by Shacham [70], named ***Type 4***, where one can both hash onto group G_2 and apply efficiently computable homomorphism from group G_2 to G_1[8].

There are two problems with this new pairing type. One is the inefficient hashing onto second input group G_2 and the second one is the vulnerability introduced into the original scheme proposed in [58] where revocation checking algorithm may falsely accept signatures generated by revoked group members. Chatterjee et al. [73] proposed a fix to this security problem and give an efficient algorithm for hashing onto group G_2.

So, one must be careful while designing cryptographic schemes that are based on pairings and have in mind that there is no known pairing type which satisfies the following three properties at the same time[9].

1. Both input groups $G_1,\ G_2$ are cyclic,

2. One can hash strings to both input groups G_1 and G_2 of which the discrete logarithm to a fixed base is unknown,

3. There is an efficiently computable homomorphism $\psi : G_2 \rightarrow G_1$, however, there is no efficiently computable one in the reverse direction, $\psi' : G_1 \rightarrow G_2$.

3.2.2 Hardness Assumptions in Pairing-based Cryptography

Bilinear hard problems are applicable to the groups over which an efficient and non-degenerate bilinear pairing can be defined. For the following number theoretic problems, it is assumed that G_1 and G_2 are groups of same prime order q generated by g_1 and g_2,

[8] Definitions for the pairing types including the fourth type are given in [72].

[9] In [74], it is argued that any two of these three properties are satisfied, but not all of them.

($G_1 = \langle g_1 \rangle,\ G_2 = \langle g_2 \rangle$), respectively. Furthermore, G_M is assumed to be a multiplicative group of order q. It is also assumed that an efficiently computable, non-degenerate bilinear pairing e exists such that $e : G_1 \times G_2 \rightarrow G_M$.

Definition 18 ***Bilinear Diffie-Hellman Problem :*** *Given $g_i^a,\ g_j^b,\ g_k^c$, compute $e(g_1, g_2)^{abc}$ where $i, j, k \in \{1, 2\}$.*

So, there are four different problems stated as $BDHP_{ijk}$, corresponding to (i, j, k) $\in$ {(1,1,1), (1,1,2), (1,2,2), (2,2,2)}. While for Type 1 pairings, these four are all the same, they are all different for a Type 3 pairing. For Type 2 pairings, problems with more input points chosen from G_2 are no harder than the ones having more inputs from G_1. BDH assumption states that it is computationally infeasible for an adversary to solve this problem and it was first used by Joux [54] and Sakai et al. [57] without stating this fact explicitly. BDHP under Type 1 pairing is utilized by Boneh and Franklin [55] to derive the well-known identity based encryption scheme.

Definition 19 ***Decisional Bilinear Diffie-Hellman Problem :*** *Given g_i^a, g_j^b, g_k^c, $e(g_1, g_2)^z$, decide if $z = abc$ where $i, j, k \in \{1, 2\}$.*

There are again four different problems for the previously stated combinations of the input groups and the previous discussion also holds for this problem.

Definition 20 ***q-Strong Diffie-Hellman (q-SDH) Problem :*** *Given $(q + 2)$-tuple (g_1, g_2, g_2^{γ}, $g_2^{(\gamma^2)}$, ..., $g_2^{(\gamma^q)}$) as input, where there exists an efficiently computable homomorphism, $\psi(g_2) = g_1$, output a pair ($g_1^{1/(\gamma+x)}, x$), where $x \in \mathbb{Z}_p^* \backslash \{-\gamma\}$.*

q-SDH problem is first introduced and used by Boneh and Boyen [75] and proven to be held in generic groups. Then, it is redefined by Boneh and Boyen [76] which is supposed to be more secure as follows;

Definition 21 ***q-SDH Problem definition of [76] :*** *Given $(q + 3)$-tuple (g_1, g_1^{γ}, $g_1^{(\gamma^2)}$, ..., $g_1^{(\gamma^q)}$, g_2, g_2^{γ}) as input, where there exists $\psi(g_2) = g_1$, output a pair ($g_1^{1/(\gamma+x)}, x$), where $x \in \mathbb{Z}_p^* \backslash \{-\gamma\}$.*

The following two definitions are the pairing based hard problems based on the original LRSW problem (cf. Definition 7) defined over finite field multiplicative groups;

Definition 22 ***Bilinear LRSW Problem (BLRSW) :*** *Let $G_1 = \langle P_1 \rangle$ and $G_2 = \langle P_2 \rangle$ be cyclic groups of prime order q and let $X \in G_1$, $Y \in G_2$ where $X = xP_1$ and $Y = yP_2$. Assume that there exists an oracle that, on input of a value $f \in \mathbb{Z}_q$, outputs a triple, $\sigma = (A,\ B,\ C) = (A,\ yA,\ x + fxyA)$ where $A = zP_1$ for a randomly chosen $z \in \mathbb{Z}_q$. Then, produce such a triple for value f' which is not queried to the oracle.*

Definition 23 ***Blind Bilinear LRSW Problem :*** *Let $G_1 = \langle P_1 \rangle$ and $G_2 = \langle P_2 \rangle$ be cyclic groups of prime order q and let $X \in G_1$, $Y \in G_2$ where $X = xP_1$ and $Y = yP_2$. Assume that there exists an oracle that, on input of a value $F = fP_1$ where $f \in \mathbb{Z}_q^*$, outputs a triple, $\sigma = (A,\ B,\ C) = (A,\ yA,\ x + fxyA)$ where $A = zP_1$ for a randomly chosen $z \in \mathbb{Z}_q$. Then, produce such a triple for value $F' = f'P_1$ which is not queried to the oracle.*

Blind BLRSW problem is no easier to solve than the original BLRSW problem, perhaps harder.

3.2.3 Pairing Implementations

As mentioned previously, pairings are first used to attack elliptic curves where Weil and Tate pairings are used in that purpose [51, 52]. After realization of the beneficial properties of pairings for cryptographic usage, researchers start searching for efficient algorithms for the computation of these pairings in order to attain practical implementations. As a result, more efficient pairing implementations, called Eta and Ate pairings and their generalizations, are developed (cf. [77, 78, 79]). These pairing implementations reduce the cost of pairing operation which is the main obstacle to the creation of efficient pairing-based cryptographic schemes.

In the following we briefly describe these pairings, but first we give the necessary definitions.

Definition 24 ***Cofactor :*** *Let $E(\mathbb{F}_q)$ be an elliptic curve defined over finite field $\mathbb{F}_q$ and let G be a subgroup of $E(\mathbb{F}_q)$ with order r. Then,* ***cofactor*** *of group G is denoted by h*[10] *where $h = \#E(\mathbb{F}_q)/r$.*

Definition 25 ***Embedding degree, k, of an elliptic curve :*** *Let $E(\mathbb{F}_q)$ be an elliptic curve defined over $\mathbb{F}_q$ and $P \in E(\mathbb{F}_q)$ be a point of prime order r. If q and r are coprimes, then the embedding degree of point P is the smallest positive integer k such that $r \mid q^k - 1$.*

Then, μ_r[11] denotes the algebraic cyclic[12] group of *r-th roots of unity* in $\mathbb{F}_{q^k}^*$, where $\mathbb{F}_{q^k}^*$ is the smallest extension of $\mathbb{F}_q$ containing all the r-th roots of unity.

An *r-torsion point* P is a point whose order divides r (either r or any factor of r), that is $rP = \mathcal{O}$. Let $E(\mathbb{F}_q)[r]$ denote the set of r-torsion points in $E(\mathbb{F}_q)$, and $E[r]$ (or equivalently $E(\mathbb{F}_{q^k})[r]$) denote the set of all r-torsion points which is a subset of $\overline{\mathbb{F}}_q$ and is isomorphic to $Z_r \times Z_r$.

Interested reader may refer to [43, 45, 77, 78, 80], and Chapter 3 of [74] for the theory of divisors and detailed computation of the pairings.

i. **Weil Pairing :** The Weil Pairing is introduced by Weil [81] which is applied over elliptic curves defined over a finite field $\mathbb{F}_q$ where q is a prime power. Let r and q are relatively prime numbers. Then, Weil pairing is a family of maps e_r,

$$e_r : E[r] \times E[r] \rightarrow \mu_r \tag{3.5}$$

Weil pairing, e_r, is bilinear, nondegenerate and $e_r(P, P) = 1$ for all $P \in E[r]$. Additionally, it possesses the antisymmetry property which is not present for the Tate pairing and its successors. Weil pairing can be computed in polynomial time by Miller's algorithm [82].

ii. **Tate Pairing :** Tate pairing was introduced by Tate [83] and then extended by Licht-

[10] For cryptographic purposes, smaller cofactors are preferable, i.e. $h \leq 4$. If the cofactor is 1, then elliptic curve group itself is a prime order group.

[11] $\mu_r = \left\{ x \in \mathbb{F}_{q^k}^* : x^r = 1 \right\}$.

[12] Since p, characteristic of the curve $\mathbb{F}_q$, does not divide r, solution to r-th roots of unity has no multiple roots in $\mathbb{F}_{q^k}^*$ and therefore forms a cyclic group.

enbaum [84] that enables explicit computation. Tate pairing was considered over finite fields for the first time by Frey and Rück [52, 85], which gives rise to the use of Tate pairing for cryptographic purposes. Tate pairing can be defined as follows;

Let $E(\mathbb{F}_q)$ be an elliptic curve defined over finite field $\mathbb{F}_q$ and let r be an integer coprime to q which divides #E($\mathbb{F}_q$). Then, Tate pairing is a map

$$E[r] \times E(\mathbb{F}_{q^k}) \ / \ rE(\mathbb{F}_{q^k}) \rightarrow \mathbb{F}^*_{q^k}/(\mathbb{F}^*_{q^k})^r \tag{3.6}$$

where k is the embedding degree of $E(\mathbb{F}_q)$. The output of the Tate pairing is applied a final powering by $(q^k - 1)/r$ in order to get a unique value in μ_r.

The main advantage of the Tate pairing over Weil pairing is that the second input can be any point of $E(\mathbb{F}_{q^k})$. On the other hand, the second input for the Weil pairing must be an r-torsion point.

iii. **Eta Pairing :** Eta pairing is introduced by Barreto et al. [77] as a derivative of the Tate pairing for the supersingular curves.

iv. **Ate and Twisted Ate Pairings :** Ate pairing and the counterpart of Eta pairing applied to ordinary curves called twisted Ate pairing are introduced by Hess et al. [78].

v. **Generalized variants of Eta and Ate pairings :** Generalized variants of the Eta and Ate pairings are proposed by Lee et al. [86] named as R-ate pairing, Matsuda et al. [87] named as optimized versions of Ate and twisted Ate pairings, and Zhao et al. [88], all of which shorten the loop length of the Miller's algorithm.

3.2.4 Pairing-friendly Curves

Both supersingular and ordinary elliptic curves can be used as input groups to pairings. However, symmetric pairings can be achieved only by using supersingular elliptic curves, while asymmetric pairings are defined over ordinary curves.

Although pairings can be defined over all types of curves, efficient computation of pairings require such curves to have small embedding degrees (cf. Definition 25). On the

other hand, for cryptographic purposes, these curves must also have a large prime-order subgroup in order to thwart attacks (cf. Section 3.1.3) on elliptic curve discrete logarithm type problems.

The security of pairing based cryptosystems are based on the hardness of number theoretic problems defined over both elliptic curves $E(\mathbb{F}_q)$, and finite fields $\mathbb{F}_{q^k}$. Therefore, one needs to work with a subgroup of $E(\mathbb{F}_q)$ with sufficiently large prime order r, and on a sufficiently large prime order multiplicative subgroup of extension field $\mathbb{F}_{q^k}$[13]. For example, to attain 80-bit security, one must use an elliptic curve with $r \geq 2^{160}$ and $q^k \geq 2^{1024}$.

On the other hand, from the efficiency point of view, arithmetic over the underlying field will be faster with a smaller q and transmission of elliptic curve points will require less bandwidth. Hence, one should keep q as small as possible and use larger k to achieve the desired security level.

Types of curves that have small embedding degree together with a large prime order subgroup are called ***pairing friendly elliptic curves***. Following are the well-known pairing friendly curves utilized in pairing-based cryptosystems.

3.2.4.1 Supersingular Elliptic Curves

Following is the definition of supersingular elliptic curves;

Definition 26 *(see Section IX.10 of [89]) Let E be an elliptic curve defined over field $\mathbb{F}_q$ where $q = p^m$, and p is the prime characteristic of the underlying finite field. Then, E is supersingular if one of the following conditions holds;*

1. $\#E(\mathbb{F}_q) \equiv 1$ *(mod p) which is an equivalent statement that characteristic p divides the trace of Frobenious t (cf. Theorem 11)*
2. *E has no points of order p over $\overline{\mathbb{F}}_q$*
3. *The endomorphism ring of E over $\overline{\mathbb{F}}_q$ is non-commutative.*

[13] Since r divides $q^k - 1$, k is the order of q modulo r, therefore k divides $\phi(r)$. And if r is a prime, then k divides $(r-1)$. If r is a large divisor of #E($\mathbb{F}_q$), then k is usually very large ($\approx r$) (if q is prime then we call q as r).

Corollary 27 *For supersingular elliptic curves defined over a prime field $\mathbb{F}_p$, where $p > 3$, trace of Frobenious, t must be zero.*[14]

Considering theorems 13 and 14, in the following we give possible embedding degrees for supersingular elliptic curves together with corresponding group structures;

Theorem 28 *Let $E(\mathbb{F}_q)$ be a supersingular elliptic curve, k is embedding degree, and p is the characteristic of the underlying field, then*

1. $t^2 = 0$ *and* $q \not\equiv 3$ *(mod 4), then* $k = 2$ *and* $E(\mathbb{F}_q)$ *is cyclic,*
2. $t^2 = 0$ *and* $q \equiv 3$ *(mod 4), then* $k = 2$ *and either* $E(\mathbb{F}_q)$ *is cyclic or* $E(\mathbb{F}_q) \cong \mathbb{Z}_{(q+1)/2} \oplus \mathbb{Z}_2$*,*
3. $t^2 = q$ *and* m *is even, then* $k = 3$ *and* $E(\mathbb{F}_q)$ *is cyclic,*
4. $t^2 = 2q$ *and* $p = 2$*, then* $k = 4$ *and* $E(\mathbb{F}_q)$ *is cyclic,*
5. $t^2 = 3q$*,* $p = 3$ *and* m *is odd, then* $k = 6$ *and* $E(\mathbb{F}_q)$ *is cyclic,*
6. $t^2 = 4q$ *and* m *is even, then* $k = 1$ *and* $E(\mathbb{F}_q) \cong \mathbb{Z}_{(\sqrt{q})-1} \oplus \mathbb{Z}_{(\sqrt{q})-1}$ *if* $t = 2\sqrt{q}$*, or* $E(\mathbb{F}_q) \cong \mathbb{Z}_{(\sqrt{q})+1} \oplus \mathbb{Z}_{(\sqrt{q})+1}$ *if* $t = -2\sqrt{q}$

Theorem 29 (MOV [51]) *Supersingular elliptic curves have embedding degree $k \leq 6$.*

Hence, supersingular curves in pairings imply a maximum embedding degree of 6. Furthermore, embedding degrees 4 and 6 require curve characteristics to be 2 and 3, respectively. However, there are specialized attacks on low-characteristic curves such as the one presented by Coppersmith [90] (also see [91, 92, 93]). As a result, in order to obtain higher levels of security, e.g. 256-bit, efficiently, one may need higher embedding degrees than the maximum attainable value of 6 that can be obtained from supersingular elliptic curves. Therefore, ordinary elliptic curves are preferable for the applications that necessitate higher security levels.

[14] For a supersingular curve, t ≡ 0 (mod p) implies that $|t| \geq p$. Then from Hasse's theorem (cf. Theorem 11), we know that $t \leq 2\sqrt{p}$, which can be stated as $t^2 \leq 4p$. Combining these two gives us $p \leq t^2 \leq 4p$ which implies that $p \leq 4$. Consequently, for a supersingular elliptic curve of prime order $p > 3$, $t = 0$

One subject to be mentioned is the problem of trivial solutions, $e(P,P)=1$, resulting from the direct application of pairings to points that are linearly dependent. Consider the Tate pairing where if $k>1$ and $P \in E(\mathbb{F}_q)[r]$, then $e(P,P)^{q^k-1/r}=1$.

In order to remove the linear dependency, one needs an endomorphism ϕ on $E(\mathbb{F}_q)$ such that $\phi(P) \notin E(\mathbb{F}_q)$. Application of this endomorphism to one of the input points of the Tate pairing makes the corresponding result non-trivial, $e(P,\phi(P))^{q^k-1/r} \neq 1$. Then, we should redefine the Tate pairing as modified Tate pairing (one may define modified Weil pairing in a similar manner) as $\hat{e}(P,Q)=e(P,\phi(Q))^{q^k-1/r}$. This problem is overcome by endomorphisms called Distortion Maps, which are introduced by Verheul [94] and exist only for supersingular elliptic curves. These maps take an r-torsion point and maps it into another one.

Distortion maps are not available for ordinary elliptic curves, and due to this fact, in order to utilize the ordinary curves, one must relinquish from using linearly dependent points in pairing-based applications.

In Table 3.1, supersingular curves that are suitable for pairing-based cryptosystems along with their corresponding distortion maps are given, [89, 94];

3.2.4.2 Ordinary Curves

Curves that are not supersingular are called ordinary curves. Although ordinary elliptic curves can be preferred over supersingular curves in pairing-based applications that necessitate higher levels of security, e.g. 256-bit, they poses important problems. First of all, ordinary curves do not have distortion maps which provide eligible solution to the problem of trivial result of the pairing. In addition, ordinary curves with small embedding degrees are very rare and special constructions are required to obtain a useful one.

First problem is solved by the trace maps, but this necessitates non-optimal choice of the second input group leading to an inefficient pairing calculations. Besides, in order to overcome the second problem extensive research have been conducted to find pairing friendly ordinary elliptic curves. Consequently, ordinary curves that can be utilized in pairing based cryptosystems are first proposed by Miyaji et al. [2], named MNT curves. However, MNT curves have embedding degrees of 3, 4 and 6 and thus similar to su-

Field	Elliptic Curve	Condition	k^*	Distortion Map	Group Order
$\mathbb{F}_p$	$y^2 = x^3 + ax$	$p \equiv 3 \pmod 4$ p is prime	2	$(x, y) \mapsto (-x,\ iy)$ $i^2 = -1$	$p + 1$
$\mathbb{F}_p$	$y^2 = x^3 + b$	$p \equiv 2 \pmod 3$ p is prime	2	$(x, y) \mapsto (jx,\ y)$ $j^3 = 1$ and $j \neq 1$	$p + 1$
$\mathbb{F}_{p^2}$	$y^2 = x^3 + b$ $b \notin \mathbb{F}_p$	$p \equiv 5 \pmod 6$ p is prime	3	$(x, y) \mapsto (x^p/\alpha b^{(p-2)/3},\ y^p/b^{(p-1)/2})$ $\alpha \in \mathbb{F}_{p^6}$ with $\alpha^3 = b$	$p^2 - p + 1$
$\mathbb{F}_{2^d}$	$y^2 + y = x^3 + x + c$ $c = 0, 1$	d is odd	4	$(x, y) \mapsto (\alpha^2 x + \beta^2,\ y + \alpha^2\beta x + \beta)$ $\alpha \in \mathbb{F}_{2^2},\ \alpha^2 + \alpha + 1 = 0$ $\beta \in \mathbb{F}_{2^4},\ \beta^2 + (\alpha + 1)\beta + 1 = 0$	$2^d + 1 \pm 2^{(d+1)/2}$
$\mathbb{F}_{3^d}$	$y^2 = x^3 + 2x + c$ $c = \pm 1$	$d \equiv \pm 1 \pmod{12}$ $d \equiv \pm 5 \pmod{12}$	6	$(x, y) \mapsto (-x + \beta,\ iy)$ $i^2 = -1$ $\beta^3 + 2\beta + 2d = 0$	$3^d + 1 \pm 3^{(d+1)/2}$

Table 3.1: Supersingular curves and their Distortion maps,(*embedding degree, security multiplier)

persingular curves they are also bounded by the maximum attainable embedding degree of 6. After this initial proposal, several other types of curves with differing embedding degrees have been proposed [95, 96, 97, 98], of which the curves described by Barreto and Naehrig [96] are the most attractive ones since they provide prime order curves with embedding degree of 12, if one both needs higher levels of security and efficient implementation.

In the following we discuss some of the well-known pairing friendly ordinary curves[15];

(a) **MNT Curves** Following theorem is due to Miyaji et al. [2],

Theorem 30 *Let E be an ordinary elliptic curve defined over $\mathbb{F}_q$ such that order of the curve $n = \#E(\mathbb{F}_q)$ = $q + 1 - t$ is prime. Then, following is the characterization of MNT curves for embedding degrees k = 3, 4, 6;*

k	*q*	*t*
3	$12x^2$ - 1	-1 $\pm$ 6x
4	x^2 + x + 1	-l or x + 1
6	$4x^2$ + 1	1 $\pm$ 2x

Table 3.2: Characterization of ordinary elliptic curves due to Miyaji et al. [2]

These curves are constructed via complex multiplication methodology (cf. Chapter VIII of [44]). Suitable MNT curves with respect to their discriminants of complex multiplication can be found in Section 2.3.5 of Shacham [70]. The major downside of MNT curves is that only few values of x will generate suitable curves. After the first proposal, which fixed the cofactor (cf. Definition 24) to 1, Scott and Barreto [99] and Galbraith et al. [100] extended the MNT method by choosing a small constant cofactor other than 1 for generating more suitable MNT curves. A comparison of MNT curves and supersingular curves can be found in [101].

(b) **Freeman Curves**

[15] One may profitably refer to Freeman et al. [98] for in-depth discussion of pairing friendly curves.

Freeman [102] gives a family of curves with embedding degree 10. One can refer to Section 5.3 of [98] and Section 4.15 of [74] for detailed information on Freeman's construction.

(c) **Barreto-Naehrig Curves**

In [96], Barreto and Naehrig presented a simple algorithm for constructing elliptic curves of prime order with embedding degree of 12. This filled the gap via providing pairing friendly elliptic curves that can be implemented efficiently and utilized to develop applications demanding high security levels, i.e. 128-bit or more.

Their algorithm takes the desired security level, that is the order of the curve in bits, and outputs parameters p, n, b, and y' such that the curve $y^2 = x^3 + b$ has prime order n over finite field $\mathbb{F}_p$, and generator $P = (1, y')$ of the curve with the following parameterizations;

$$t = 6x^2 + 1$$

$$n = 36x^4 + 36x^3 + 18x^2 + 6x + 1$$

$$p = 36x^4 + 36x^3 + 24x^2 + 6x + 1$$

where x may take both positive and negative values.

They also presented both point and pairing compressions up to sixfold[16], which makes their construction especially valuable for the applications with low bandwidth requirements.

[16]Pairing compression can be achieved for the schemes that do not require further processing of the pairing result.

Chapter 4

Group Signatures and Attestation Schemes

4.1 Introduction to Group Signatures

The concept of group signatures was introduced by Chaum and van Heyst [9]. In their setting, group entities are comprised of a number of group members and a group manager, in which any member can sign a message (or a document) on behalf of the group anonymously. Hence, anyone, within the group or outside, who received a message-signature pair can be assured that the signature is generated by a valid group member but is not able to identify the generator of that signature, and even cannot tell whether any given two or more signatures are generated by the same group member or not. But, in case of a dispute, no one but the group manager[1] has the capability of 'Open'ing a valid group signature and thus reveal the identity of the originator.

Group signatures are especially attractive for applications demanding protection of user privacy and where the organizational structure needs to be concealed such as;

- Trusted Computing [103]
- Banking (Electronic cash [39], stock or bond issuance where banks form a group of cash (stock, bond, etc.)-issuers)
- Electronic Voting and Auctions
- Government and military

[1] Revocation Manager or Opener is also used for naming the authority possessing the opening capability.

- Press releases requiring anonymity
- Identification as a group member (to get access to a restricted area) [40]

A popular example in the literature is the utilization of group signatures in invitation to submit tenders [104, 105, 106]. All companies that are to be involved in a tender form a group, and companies submit tenders anonymously using group signatures. They are all bound to their submitted tenders by anonymous signatures provided, among which the selected member's signature can be opened, thus the generator can be identified without the need for the involvement of the chosen group member. Consequently, issuer of the preferred tender will be revealed by the group manager whereas rest of companies still remain anonymous.

4.2 Properties of the Group Signature Schemes

A group signature scheme can be defined as follows;

Definition 31 *A group signature scheme is a digital signature scheme with following procedures;*

- **SETUP:** If there is a single entity (group manager) involved in registration of members as well as opening of the signatures, then this is a probabilistic algorithm, given security parameter k as input, generates the group public key, group manager's secret-public key pairs to be used in registration protocol, the opening key to be used in revealing the originator of a given signature and all other necessary system parameters. If the group is a static group[2], this algorithm also generates and distributes the group members' secret keys (and their corresponding certificates) where the number of group members are predetermined.

  Otherwise, if group manager's role is shared among two distinct entities[3], then this is an interactive protocol between a group manager (issuer), revocation manager

[2]The number and identities of members are decided in setup phase and new members cannot be added or removed later on [10].

[3]Issuer [10] which is responsible for generation of the membership keys (or credentials) and Opener (or Revocation manager).

(opener), and respective group members, in case of a static group, that generates all necessary public and private keys together with the required system parameters.

- **JOIN:** In case of a dynamic group[4], in which group members can be added to and/or removed from the group, this is an interactive protocol between the group manager (or Issuer) and a user, that provides the user with a secret key (and a certificate on this key), hence results in user becoming a new member of the group.

- **EVOLVE**[5]**:** An algorithm, given a valid group member's signing key for time period i as input, outputs the corresponding signing key for the subsequent time period $i + 1$. This procedure is used in forward secure group signature schemes providing revocation of group members and is first defined and used by Song [107].

- **SIGN:** A probabilistic interactive protocol between a group member and a user, being either a group member or an outsider, whereby a group signature is computed on a given message *m* by the group member's secret signing key, which can be verified by anyone with the group public key.

- **VERIFY:** An interactive protocol between a group member and a user (verifier), upon which the validity of a given signature is determined by means of a group public key and the signed message.

- **OPEN:** Given a signature on a message along with the message itself, this procedure reveals deterministically the identity of the signer using the revocation manager's[6] opening secret key.

A group signature scheme should provide the following security properties;

– **CORRECTNESS:** Any group signature produced by an authorized group member via sign procedure must be valid and accepted by the corresponding verification

[4]The number of group members and their respective identities are not known in the setup phase, in the sense that an entity can join the group and obtain his secret signing key at any time via an appropriate registration protocol [11].

[5]General definition of group signatures does not involve this procedure, and it is included here for being comprehensive.

[6]Or the group manager's key in case there is a single group authority.

procedure. In addition, opening algorithm correctly recovers the identity of the originator of a given valid signature.

– **UNFORGEABILITY:** Only registered group members are able to sign messages on behalf of the group. For any user outside the group, it is computationally infeasible to produce such a signature that is accepted by the verification algorithm.

– **ANONYMITY (UNTRACEABILITY):** Given a valid message-signature pair, identifying the corresponding signer is computationally infeasible for anyone but the group/revocation manager.

– **UNLINKABILITY:** Given a list of signatures, it is computationally infeasible to decide whether any two of these signatures are generated by the same group member or not.

– **EXCULPABILITY**[7]**:** No entity within the group, either the group members or the group manager, is not able to produce signatures on behalf of the other group members.

– **TRACEABILITY:** A valid signature that is generated by a registered group member can be opened and hence the corresponding user can be identified correctly by the group/revocation manager.

– **COALITION-RESISTANCE (UNAVOIDABLE TRACEABILITY):** No coalition of group members, even if all group members collude, cannot generate a valid group signature which cannot be traced to any one of the group members by the group manager via the opening key. This requirement was first stated explicitly by Ateniese et al. [105][8] and separated from the traceability property.

– **NON-FRAMING:** A coalition of group members combining their secret signing keys is not able to generate a valid signature that the opening algorithm traces it

[7] Exculpability is introduced by Ateniese and Tsudik [108].

[8] The first group signature scheme that is provably secure against coalition-resistance is also presented in this work.

to an authorized group member who is outside this coalition. Non-Framing is first considered in [104] and it is a version of the coalition-resistance property.

– **REVOCABILITY**[9]**:** A group signature produced by a revoked member via sign algorithm must be rejected by the verification algorithm. On the other hand, if a member is not revoked, then the correctness property must hold for the member's signatures. This is actually an optional property which is satisfied in group signature schemes that are designed to allow for the removal of the group members.

Formal definitions for the security properties of group signatures mentioned thus far[10], together with the attacker capabilities, are first given by Bellare et al. [10] for static groups. Later on formal definitions are given for dynamic group by Bellare et al. [11], in which informal properties stated in previous works are combined into three comprehensive security requirements; anonymity, traceability and non-frameability. In their work, framing and exculpability are implied by the non-frameability. Coalition-resistance and unforgeability requirements follow from the traceability together with the non-frameability, whereas traceability is implied solely by the traceability property. Anonymity and unlinkability properties are covered by anonymity.

4.3 Evolution of Group Signatures

In their seminal paper, Chaum and van Heyst [9] describe the group signature concept and give four different realizations, in one of which the anonymity is preserved unconditionally. On the other hand, it is protected computationally in the rest of the realizations based on either the difficulty of factoring or computing the discrete logarithms. Regarding the schemes providing computational anonymity, in two of them, the addition of new members to the group is not allowed. In both of them, in order to obtain the identity of a signer or to open a signature, group manager needs to contact each one of the group members.

[9] Explicitly stated first by Ateniese et al. [106].

[10] Except the revocability requirement which is considered neither in static [10] nor in dynamic [11] versions.

In both schemes, the size of the group public key is linear in the number of group members which makes them inefficient. In any case, distributing the group manager's role to more than one entity cannot be provided by either of these proposed schemes and it is left as an open problem.

After the initial work of Chaum and van Heyst, numerous group signature schemes have been proposed with the intent to improve both efficiency and security of the proposed schemes. In [104], Chen and Pedersen addressed the problem of distributing the group manager's role as well as the new group member addition problem.

Former problem is overcome via providing an auxiliary information which can be shared among a subset of the group in interest by utilizing the non-interactive and verifiable secret sharing scheme of [109]. As a result, members in this subset together can identify the user without the need of a single group manager. Along with that, they also solve the problem of the group manager contacting each group member to open a given signature by utilizing this auxiliary information provided by each signing member. This is realized by so-called double-signing method, in which each group member has two secret signing keys, one is known only by the group member herself, and the other one is used as an auxiliary information known also by the group manager (or shared among a predetermined subset of group members).

Their group signatures are based on undeniable signatures introduced by Chaum and Antwerpen [110] and used a protocol that proves the knowledge of one secret key (membership key of the prover) out of many (all membership keys).

However, their proposals are also inefficient in a way that the size of the group public key is also linear in the number of group members, and as mentioned by the same authors in subsequent works [111, 112], group manager can falsely accuse a group member of signing a particular message with the help of auxiliary secret signing key handed over the group manager by the group member during registration. In these works, authors proposed a scheme providing unconditional security against framing which cannot be obtained by the previous scheme [104] and they have also stated that for the schemes providing information-theoretic anonymity, the length of secret keys and auxiliary information increase linearly with the number of group members and in the number of

signatures allowed to be generated by each member.

Therefore, in order to develop a practical and implementable group signature scheme, one has to give up on unconditional anonymity and try to find schemes that provide computational anonymity which can be attractive for real-life applications.

Camenisch [38] presented a more efficient group signature scheme in terms of the cost of signature computation and the length of the group signature generated, which provides computational anonymity where opening is independent of the number of group members. But, again the size of the group public key as well as the signature depends on the number of group members. Building blocks of the group signature schemes presented comprise a variant of ElGamal encryption [7], secret sharing scheme of Shamir [113] which is used in constructing the generalized group signature scheme, signature knowledges of a discrete logarithm (cf. Section 2.3-1), equality of discrete logarithms with respect to different bases (cf. Section 2.3-2) and a representation (cf. Section 2.3-5).

Basic scheme presented in Camenisch [38] allows for the addition of group members dynamically after the initial setup, and it can also be generalized in a way that a subset of authorized group members can sign on behalf of the group acting like a single signer. Both schemes allow sharing of the group manager's functionality utilizing secret sharing schemes of Shamir [113] and Feldman [114].

All of the group signature schemes mentioned so far, [9, 38, 104, 111, 112], have the following important drawbacks;

- The length/size of the group public key and/or group signature depends on the number of group members.

- Addition of new members to the group requires either modification of the group public key along with the generation and distribution of new secret signing keys to all members or restarting the whole system.

- Revocation of group members can be performed only by revoking all the members and then reissuing secret signing keys to all members with a corresponding change in the group public key.

Schemes presented in [115, 116] possess the fixed size public keys but were shown to be flawed in [117, 118, 119].

The state-of-the-art in the field of group signatures is presented by Camenisch and Stadler [1], which addressed all of these common shortcomings of the previous group signature schemes except the revocation mechanism. This is accomplished with increased cost of computations required for the generation and verification of the group signatures, although these computations are independent of the group size.

In order to realize such a scheme, authors utilize novel techniques such as signature knowledges of double discrete logarithms (cf. Section 2.3-6), *e-th* root of discrete logarithms (cf. Section 2.3-7) and *e-th* root of components of representations (cf. Section 2.3-7&5), all of which are secured in the random oracle model [29, 30, 31]. They base the security of their group signature scheme on newly introduced computational problems that are assumed to be hard, i.e. double discrete logarithm and root of discrete logarithm problems (cf. Definition 4&5).

Dependence of the group public key length and signature size on the number of group members is prevented via employing membership certificate. In this respect, along with the group manager's public-private key pair required for encryption, a signature key pair is also generated which is used to create certificates for secret signing keys of the group members. Since this signature key pair is generated independently from the group members, verification of a credential can be performed without referring to any one of the group members.

Additionally, the separation of membership management (issuance of membership certificates) and revocation management (identification of the originator of a given signature) is stated explicitly. These roles can also be shared among more than one entity to provide protection against dishonest group membership and opening/revocation managers.

4.3.1 Group Signature Approach of Camenisch and Stadler [1]

The approach behind the group signature scheme presented by Camenisch and Stadler [1] can be summarized as follows; The group manager computes two key pairs, one for an or-

dinary digital signature scheme, (s_M, p_M) used in generation of membership certificates, and the other for a probabilistic encryption scheme, (e_M, d_M), which is required for the identification and/or revocation of the dishonest group members.

A user, in order to *join* group, first selects a secret signing key x randomly, and then computes corresponding membership key $z = f(x)$ where f is a suitable one-way function. User, then commits on value z, by signing it, and sends it to the group manager who computes corresponding membership certificate $v = sign_{s_M}(z)$ and sends it back to the user. As a result, user becomes a group member and sets his membership key as (x, z, v).

This group member *sign*s a given message m, by first encrypting message-membership key pair with a probabilistic encryption scheme utilizing a random value r into ciphertext $c = enc_{e_M}(m, z, r)$, and then proving the knowledge of secret values x and v along with a proof that the encryption is performed on z and m using r.

Opening of the signature is performed by the group manager by decrypting the received ciphertext c and obtaining the membership key, and thus the identity of the signer. To assure that the identity of the signer is actually the one revealed by the group manager, group manager discloses the value z and member's corresponding commitment to it together with a proof that the decryption of c results in the given message-membership key pair.

Although the scheme presented by Camenisch and Stadler [1] removed one of the most important barriers that hinders the deployment of group signature in real-world applications, there exist subtle problems. Most important ones, also stated by Ateniese and Tsudik [108], are the lack of coalition resistance[11,12] and the lack of efficient revocation mechanism.

[11] Applications where coalition resistance is not required are limited in the sense that, in those kind of applications group members must be reluctant to share their secrets with other group members, i.e. electronic lotteries.

[12] An attack is presented by Ateniese and Tsudik [108] against coalition resistance of the basic group signature scheme of [1], along with the proposed fixes which are not proven to be secure.

4.3.2 Provably Secure Group Signatures against Coalition Attacks

The state-of-the-art group signature scheme which provides provable security based on cryptographic assumptions is proposed in [25, 26] which has not been accomplished in any of the previous works.

Efficient group signature schemes proposed so far [1, 120] put forth the idea of generating group signatures by making use of two ordinary digital signature schemes along with a probabilistic semantically secure encryption scheme [121, 122]. One of these signature schemes is used to create certificates for the secret signing keys of the authorized members, and the other one is used to create actual group signatures by group members. This separation is analyzed in a comprehensive manner in [123]. Again, an encryption scheme is required for the opening.

In order to attack against coalition resistance of a group signature scheme, the attacker must try to compromise the signature scheme used to grant membership credentials to group members in the registration phase. This kind of an attack can be seen as an adaptive chosen message attack against the join protocol where the attacker has the capability of querying a join oracle with the member secrets and obtaining corresponding credentials of his choice except the one being attacked.

In [25, 26], coalition resistance requirement is satisfied via utilizing a new number-theoretic assumption which is a variant of strong-RSA assumption called *modified strong-RSA assumption* together with the discrete logarithm and DDH assumptions, (cf. Definitions 9, 10, 1 and 3). The same idea is employed in these works as the one put forth in [1] (cf. Section 4.3.1), but now based on newly introduced number theoretic assumption. Building blocks for the scheme are four signature proofs of knowledge which can also be combined [120], namely signature proofs of knowledge of discrete logarithm, equality of discrete logarithms, one out of two discrete logarithms and signature proof of knowledge that a discrete logarithm lies within a certain interval (cf. Section 2.3 and [26]).

After the state-of-the-art proposal, a new group signature scheme which improves the first coalition resistant scheme is introduced by Ateniese et al. [105][13] which is based

[13] which will be denoted by [ACJT] from now on.

on the original strong-RSA assumption (cf. Definition 9). In [ACJT], improvements were made on the efficiency and security of the join protocol which is statistically zero-knowledge with respect to the secret key of the member. This is not provided in the previous scheme in which the group member must provide an inefficient proof that some number is the product of two primes. This product is composed of one random prime and a prime of special form (the secret key of the member), which is susceptible to Coppersmith's attack [124].

Coalition resistance of the protocol is based on the following theorem;

Theorem 32 ***Coalition-resistance (cf. Section 6 of [105]) :*** *Under strong-RSA assumption, a group certificate [A_i=$(a^{x_i}a_0)^{1/e_i} \pmod n$, e_i] with $x_i \in \Lambda$ and $e_i \in \Gamma$ can be generated only by the group manager provided that the number K of certificates the group manager issues is polynomially bounded, where $\Lambda =]2^{\lambda_1} - 2^{\lambda_2}, 2^{\lambda_1} + 2^{\lambda_2}[$, $\Gamma =]2^{\gamma_1} - 2^{\gamma_2}, 2^{\gamma_1} + 2^{\gamma_2}[$ and $\lambda_1 > \epsilon(\lambda_2 + k) + 2$, $\lambda_2 > 4l_p$, $\gamma_1 > \epsilon(\gamma_2 + k) + 2$, $\gamma_2 > \gamma_1 + 2$ and $\epsilon > 1$, k and l_p are security parameters. Here, ϵ controls the tightness of the statistical zero-knowledgeness, and parameter l_p sets the size of the modulus to use.*

An improvement was made by Camenisch and Groth [125] over the original [ACJT] scheme. In their work, signature scheme proposed by Camenisch and Lysyanskaya [126] is employed in credential generation process due to the fact that it provides efficient protocols to prove the knowledge of such a signature, which is a major requirement for an efficient group signature scheme.

The basic group signature scheme presented, which also allows dynamic member addition, has full-anonymity and full-traceability according to Bellare et al. [10] terminology. Furthermore, it is nearly 20 times efficient than the state-of-the-art [ACJT] scheme and the security is proven under the strong-RSA and DDH assumptions in the random oracle model. Basic scheme can be easily extended to support revocation, and extended scheme is much more efficient than other proposed extensions that add revocation capability [106, 127] to the original [ACJT] scheme.

In addition, in the full version of Camenisch and Groth [125], given reference to Hansen and Pagels [128], it is argued that signature generation and verification is computationally

faster than any one of the pairing-based signatures proposed in [59, 60, 129]. However, for the same level of security, shorter signature sizes are obtained by pairing-based signature schemes (cf. Section 4.5) due to the short representation of group elements.

4.4 Revocation in Group Signatures

In order to withstand the demand of practical usage and rapid deployment of group signatures, proposed schemes should be efficient and dynamic in nature, supporting both inclusion and deletion of group members. Although group signature schemes presented in previous sections (such as [1, 25, 26, 105]) support efficient member addition without a need for a change in the group public key and for the reissuance of certificates for already registered group members, they do not provide a viable solution for member deletion.

In group signature settings, anonymity and unlinkability properties that are provided to the users can easily be abused by malicious users. Those users presumed to be guilty must be identified and prevented from generating valid signatures on behalf of the group, thus must be revoked by an efficient and secure mechanism. Another important problem is the backward linkability of past signatures generated by a revoked user. That is to say, an efficient mechanism should be devised for member revocation such that anonymity and unlinkability of the signatures originated from the non-revoked members as well as the past signatures of a revoked user should remain intact. As a result, anyone can easily authenticate a valid signature that is produced by a non-revoked user in an efficient and public manner and without the ability to find out secret information, such as one that can help one to link signatures generated by the same user. This should be performed by anyone without the need for the group manager. Additionally, in case of a legal dispute, opening of the group signatures by a designated group authority should also be provided.

The revocation of credentials has been a difficult task in public key cryptosystems where the public keys of the members must be authenticated by other users via the credentials given on member public keys granted by a trusted center. In group signature schemes, it is more complicated to implement such revocation for the group member credentials. This complication was first mentioned in [108] where authors presented two

generic solutions;

(a) **Certificate Revocation Lists (CRLs) :** CRLs are updated and broadcasted periodically by the group manager which is composed of the list of identities of the group members. CRL-based revocation is attractive due to the fact that signing group member does not have to possess this list and be aware of any changes made to the list. In addition, although the signer must prove in a way that he is not listed in CRL, computations for revocation checking are placed on the verifiers' side which are generally more powerful than the signing entities.

    Two questions arise against the usage of the CRLs. One has to do with identifying the group member: Since group signatures are anonymous and unlinkable then it is not clear how to identify a group member. The other one has to do with the secret key exposure: Exposing a secret value and putting it onto the list breaks the rule of anonymity and unlinkability of the past signatures. This is especially important when a member's secret key is compromised by an adversary and thus needs to be revoked without giving up the anonymity and unlinkability of the past signatures. This is a case where forward secure [130, 131] group signatures may be required.

(b) **Re-issuance based Revocation :** To revoke a group member, group manager first changes the group public key and then re-issues membership certificates to all the registered members except to revoked ones. This is suitable only for small and stable (static) groups where deletion of group members rarely occurs due to its heavy computation and communication costs. To achieve this, each group member must be notified somehow of the re-issuance and participate in an interactive join protocol which results in both computational and communication burden on users.

Revocation in group signatures is first addressed explicitly by Bresson and Stern [132] where authors present revocation extension on the group signature scheme of Camenisch and Stadler [1]. Proposed scheme is based on certificate revocation lists which is composed of the membership keys of the revoked members. Signatures are generated as in the original proposal with an additional zero knowledge proof that public membership key

used in the signature is not one of the keys listed in the revocation list. In their proposal, main challenge is the proof provided by the authenticating user that the plaintext of an ElGamal encrypted value is not one of the values that are present in revocation list. In order to do so, signer provides so-called witness values[14] for each revoked membership key in the list, which is some random power of the division of membership key of authenticating user by a corresponding revoked key in the list. Along with each witness value is a proof that this witness value is well-formed such that the numerator of the given witness is the plaintext of the ElGamal encrypted value.

Since the proposed scheme exposes only public information and does not leak any secret value, it provides secure deletion of group members without compromising the principles of anonymity and unlinkability of signatures produced by valid members as well as the past signatures of these revoked members. On the other hand, signature size grows linearly with the number of revoked group members which makes this scheme impractical. Besides, the group signature scheme on which this revocation capability is built on, is not proven to be coalition resistant (cf. Section 4.3.2).

In [107], Song proposed the first forward secure group signature scheme which provides revocation capability on the provably secure group signature scheme of Ateniese et al. [105], resulting in efficient constant-length signatures.Forward secure signatures are especially important for group signatures where the impact of key exposure increases with the group size, the concept of which was first introduced by Anderson [130] for ordinary digital signatures. In signature schemes providing forward security, compromise of a group member's secret key does not give adversary the capability of forging group member's past signatures because the attacker is unable to compute valid signatures pertaining to pre-revocation period using the captured key. In order to achieve forward security, where the public key of the scheme stays fixed but group signing keys evolve (cf. Section 4.2) over time, the author borrowed and apply the techniques from [131] and [134].

Revocation in Song [107] is examined considering the following properties;

– *Public revocability :* Nobody is able to generate valid signatures using an exposed group signing key after its revocation by the group manager.

[14] The idea is borrowed from [133].

– *Retroactive public revocability :* Signatures generated by the exposed key between the period of key being stolen and the period of exposure being discovered should be verified as invalid, but signatures produced by non-revoked keys should remain valid, anonymous and unlinkable.

– *Backward Unlinkability :* Signatures generated by the exposed key before the time of exposure should be accepted as valid and remain anonymous and unlinkable.

In order to achieve retroactive public revokability together with backward unlinkability, the approach involves the following: (1) the division of the time into fixed length periods in which the group public key is valid, and then (2) making group signing keys evolve within these time periods using a suitable one-way function. Revocation is made possible by revocation tokens. Two different schemes based on [ACJT] signature scheme are proposed with differing evolve procedures and revocation tokens.

In the first scheme, squaring is used as a one-way function to achieve group signing keys to evolve, whereas in the second scheme, a deterministic one-way method is given such that with an initial random prime, a sequence of prime numbers are generated and used to evolve group signing keys. These two schemes have differing security structures and procedure performances (cf. Section 6 of [107]). But, with extra cost, *time limited revocation* can be made possible in both schemes where group signing keys are issued by the group manager in such a way that issued keys are only valid for the specified time interval.

The important drawbacks in these schemes are the predetermined number of periods where the group public key is valid, the use of fixed length time periods and the requirement for a clock synchronization among the group entities. In addition, there is no way to save the legitimate signatures generated by a revoked user before the exact time of revocation within the time period when revocation takes place. This is especially important if time period intervals are too long where backward unlinkability will not be satisfied for many such signatures produced when the user was actually legitimate. Another problem is the inefficiency of the second scheme due to the computation of predetermined number of primes that are used in key evolve procedure.

The works of Ateniese et al. [106] and Camenisch and Lysyanskaya [127] are also important studies for incorporating revocation mechanism into group signatures that lack such procedures. In both of the proposed extensions, revocation capability is added to the basic [ACJT] scheme.

Given all these efforts, revocation necessitates more research in order to develop mechanisms that satisfy the following requirements;

- Shorter CRL which is sublinear in the number of revoked members, and secure and efficient CRL update and distribution for schemes employing CRL-based revocation (cf. Section 4.4-a).
- More efficient signature generation and verification algorithms possessing revocation capability (especially procedures depending on much more efficient SPKs).
- Relaxed predetermined number of periods and length of time intervals for the schemes providing retroactive revocation.

4.5 Pairing based Group Signatures

Boneh et al. [135] give the first construction of digital signatures from bilinear pairings (cf. Section 3.2.1). Since then, pairings have been drawing increasing attention and they are used in constructing group signatures as well as direct anonymous attestation schemes (cf. Section 4.6)[15].

We can analyze pairing-based group signature schemes in two differing categories based on the security assumptions on the generation of membership certificates;

1. *Bilinear LRSW based schemes*

   LRSW signature scheme is introduced by Lysyanskaya et al, in [20] for Pseudonym systems[16]. The corresponding bilinear LRSW assumption (cf. Definition 22), utilized in pairing-based schemes, was shown to hold for generic groups and it is independent of the DDH assumption (cf. Definition 17).

[15] An application specific group signature scheme in which signer is also anonymous with respect to the group manager.

[16] Pseudonym systems were introduced by Chaum [136]

2. *Strong Diffie-Hellman based schemes*

   q-SDH assumption (cf. Definition 20) was introduced by Boneh and Boyen [75] in order to construct short signatures where security does not depend on the random oracle assumption. q-SDH has similar properties to strong-RSA assumption and may be seen as its discrete logarithm equivalent. q-SDH assumption is employed in various group signature constructions such as [3, 58, 59, 129].

The first pairing-based group signature scheme that relies on BLRSW assumption is proposed by Camenisch and Lysyanskaya [60]. They follow the same approach employed by Camenisch and Stadler [1] (cf. Section 4.3.1). In order to provide their group signature scheme with the opening capability, they make use of Cramer-Shoup (CS) encryption scheme [137] whereby membership certificates are encrypted with the public key of the group manager (or revocation manager). Since Cramer-Shoup cryptosystem is secure under the DDH assumption (cf. Definition 3), this encryption scheme is performed over the output group G_M of the selected pairing, where DDH problem is intractable.

Therefore, group signature scheme proposed by Camenisch and Lysyanskaya [60] is secure under BLRSW and DDH assumptions in G_M, since the credentials on group members' secret keys are obtained from the membership/group manager via employing the LRSW signature scheme, and opening process is realized by adapting the CS encryption scheme.

Concurrent with the Camenisch-Lysyanskaya group signature scheme, Boneh et al. [59] independently proposed a pairing based short group signature scheme based on different assumptions. First one is the q-SDH assumption, and the second one is the decision linear Diffie-Hellman assumption which is introduced in [59] and defined as follows;

Definition 33 ***Decision Linear Diffie-Hellman Assumption :*** *Let $G_1 = \langle g_1 \rangle$ be cyclic group of prime order r. Given arbitrary generators $u, v, h \in G_1$, and u^a, v^b, h^c, the probability of deciding whether $a + b = c$ or not is negligible by a polynomial time adversary. It is shown in [59] that Decision Linear Assumption holds in generic bilinear groups by presenting a lower bound on the computational complexity in the sense of Shoup [138].*

Solving decision linear problem is believed to be hard in groups where solving decision Diffie-Hellman problem is easy.

Following the introduction of decision linear assumption, authors provide a related encryption scheme based on newly introduced assumption and called it *Linear Encryption*. Since ElGamal encryption scheme is not applicable in groups where solving DDH problem is easy, as in the case of group signatures proposed under q-SDH assumption, a new encryption scheme that is secure under these settings is required in order to provide revocation manager with the opening capability.

Definition 34 ***Linear Encryption :*** *In this scheme, one randomly selects $x, y \in_R \mathbb{Z}_p$ as private keys and computes three generators $u, v, h \in G_1$ such that $u^x = v^y = h$ as the corresponding public key. Encryption of a message m is performed first by choosing random values $a, b \in_R \mathbb{Z}_p$ and then computing the ciphertext as ($A = u^a,\ B = v^b,\ C = m \cdot h^{a+b}$). To decrypt a given ciphertext ($A,\ B,\ C$), user just computes $m = C/(A^x \cdot B^y)$ and thus recovers the message.*

In group signature schemes proposed by Boneh et al. [59] and Boneh and Shacham [58], Linear Encryption is used for encrypting part of membership certificate whereby in case of a dispute, group/revocation manager opens the signature to identify the signer. In their proposals, to provide exculpability, group members participate in Join protocol where user chooses a secret y randomly and gets its membership credential (A, x, y) such that $A^{\gamma+x} \cdot h^y = g$ for some public parameter h.

After these initial proposals, Furukawa and Imai [129] and Delerablée and Pointcheval [3] proposed more efficient group signatures schemes. They both achieve this by attacking the use of linear encryption whereby part of a membership certificate is encrypted, which places more computational burden on the signer than ElGamal type encryption.

Furukawa and Imai [129] use a group G having the same order with the pairing groups $G_1,\ G_2,\ G_M$ where DDH problem is difficult to solve. Membership certificate to be encrypted is selected from this new group, and as a result, use of simpler ElGamal type encryption is allowed instead of costly Linear Encryption. They provide a comparison of their scheme with three previous proposals [59, 60, 139]. In this respect, changes are

made in the construction of Boneh et al. [59] scheme; first, a join protocol is included, and then, to make the scheme IND-CCA2 [140] secure in the non-generic model, double encryption scheme [141] variant of Linear Encryption is implemented. These changes are applied for a fair comparison since other schemes include a join protocol and they are IND-CCA2 secure.

Table 4.1 is taken directly from [129][17] which was adopted directly from Hansen and Pagels [128]. Results are obtained by assuming that the order of the groups G, G_1 and G_2 are 171 bits, therefore points in these groups are represented by 172 bits. Furthermore, points in G_M are assumed to be represented by 1020 bits.

	Variant of [59] (Sign/Verify)	*Scheme of [139] (Sign/Verify)*
# of Mult in G	-	-
# of Mult in G_1	11 / 12	20 / 13
# of Mult in G_2	0 / 2	-
# of Exp in G_M	3 / 3	6 / 2
# of pairings	0 / 1	0 / 3
Signature Size (bits)	2057	4782
Assumptions	SDH, DLDH	SDH, DBDH
	Scheme of [60] (Sign/Verify)	*Scheme of [129] (Sign/Verify)*
# of Mult in G	-	6 / 6
# of Mult in G_1	3 / 0	1 / 0
# of Mult in G_2	-	0 / 2
# of Exp in G_M	13 / 13	4 / 4
# of pairings	0 / 5	0 / 1
Signature Size (bits)	5296	1711
Assumptions	LRSW, DDH	SDH, DDH

Table 4.1: Comparison of Pairing based Group Signature Schemes

Before concluding pairing based group signature discussion, one last scheme to be mentioned is the one proposed by Delerablée and Pointcheval [3] named XSGS, eXtremely Short Group Signature. In order to avoid linear encryption, authors base security of the scheme on both q-SDH and XDH assumptions.

[17] For detailed discussion of the proposed variant of [59] and complexity related issues, refer to the original paper [129].

Definition 35 ***eXternal Diffie-Hellman Assumption :*** *XDH assumption, introduced in [142], states that, given three groups $G_1,\ G_2,\ G_M$ and a bilinear pairing $e : G_1 \times G_2 \rightarrow G_M$, solving DDH problem is easy in G_2, whereas it is hard in G_1. XDH assumption implies that there must not be an efficiently computable isomorphism from group G_1 to G_2, $\psi : G_1 \not\rightarrow G_2$.*

In Boneh et al. [59], usage of groups satisfying XDH assumption has also been suggested in order to obtain even shorter group signatures than the one originally stated without XDH assumption. Such an assumption is known to be false for supersingular curves [143] but can be implemented using MNT curves (cf. Section 3.2.4.2-a).

So, in Delerablée and Pointcheval [3], it is assumed that DDH problem is hard in group $G_1 = \langle P_1 \rangle$ and easy in group $G_2 = \langle P_2 \rangle$, under the XDH assumption which allows implementing the IND-CCA2 [140] secure ElGamal based encryption. In XSGS scheme, membership issuer has private key γ and corresponding public key $w = \gamma P_2$, whereas revocation manager has private key (ϵ_1, ϵ_2) and corresponding public key $(H = \epsilon_1 K,\ F = \epsilon_2 K)$ where $K \in G_1$. Furthermore, a secret number y is added to SDH-pair, which is known only to the user, so the membership certificate is formed as (A, x, y) where $A \in G_1$, $x, y \in Z_q$ such that $(x + \gamma)A = P_1 + yH$.

In order to sign a message, user encrypts A with the public key of the revocation manager via double ElGamal encryption and provides a signature proof of knowledge of secret values x and y in addition to the random values used in the encryption. Resulting signature consists of 4 elements from group G_1, 4 integers from Z_p and a challenge value.

In Table 4.2, we provide security assumptions made together with the computational requirements for signing and verification algorithms. In addition, corresponding signature size is given based on the same group order assumptions as the ones used while deriving results supplied in Table 4.1.

4.6 Direct Anonymous Attestation

Group signatures have been adopted in diverse application areas such as electronic cash, identity escrow, direct anonymous attestation and authentication in sophisticated access

	Scheme of [3] (Sign/Verify)
# of Mult in G	-
# of Mult in G_1	7 / 3
# of Mult in G_2	0 / 1
# of Exp in G_M	1 / 1
# of pairings	0 / 1
Signature Size (bits)	1352
Assumptions	SDH, XDH

Table 4.2: Complexity and assumptions of the scheme of [3]

control schemes [4, 16, 39, 40].

One of the advanced applications of group signatures is the Direct Anonymous Attestation (DAA) adopted by the Trusted Computing Group (TCG) [103], an initiative started[18] in order to develop standards for Trusted Computing platforms called TPM. Initial proposal for the scheme has been made by Brickell et al. [144] and it was accepted by TCG and specified in TPM specification version 1.2 [145]. Recently, this proposal was accepted as an international standard by ISO/IEC [146]. Main objective of the DAA scheme is to allow trusted computing platforms to attest themselves anonymously as being legitimate devices via a variant of group signatures.

There is a major difference between the group signatures and direct anonymous attestation schemes. In DAA schemes, opening capability of the group manager is removed and thus signatures generated by a TPM remain anonymous also to the group manager who possesses the group secret key. Consequently, the requirement of an IND-CCA2 encryption scheme in group signatures along with a protocol required to prove that a committed value is in fact contained in related ciphertext is no longer needed. Besides, in DAA, signer's role is split between two entities, namely a TPM and a Host on which the TPM resides. The intuition behind this separation is that the resource and computationally constrained TPM, which holds the secret signing key for attestation, should only perform security sensitive computations that require secret signing key and delegate other related computations to the much more powerful Host.

[18] Initially started by AMD, HP, IBM, Intel and Microsoft, and known as Trusted Computing Platform Alliance

Similar to group signature schemes, DAA proposals can be classified by the number theoretic assumptions utilized in membership certificate generation. Current proposals are based on strong-RSA [144, 147], SDH [65, 67, 68, 148, 149, 150] and LRSW [16, 66, 69, 151, 152] assumptions.

In the original proposal adopted by TCG, Camenisch-Lysyanskaya signature scheme [153] is used for credential generation, which is based both on the assumptions of strong-RSA and DDH in a finite field. Following the first proposal, Ge and Tate [147] come up with another DAA scheme, which is also based on the same assumptions, but utilizes the group signature scheme introduced by Camenisch and Michels [26] for certificate generation process.

The first time where a pairing operation is used in a DAA protocol is in the scheme developed by Brickell et al. [64, 151], which makes use of symmetric pairing operations utilizing Camenisch-Lysyanskaya signature (BCL[19]) scheme presented in [60]. Underlying BCL scheme is based on bilinear maps and the security of the scheme is proven under bilinear LRSW assumption [60], which is applicable for groups with bilinear maps.

A more efficient and scalable solution utilizing asymmetric pairings is proposed in [66], which is also based on BCL signature scheme, but adopted to asymmetric pairing setting. Asymmetric pairings are attractive due to the fact that DDH problem is believed to be hard in input groups which eliminates extra checks, and computations required for masking against DDH problem being easy in the symmetric setting. Besides, higher embedding degrees are attainable only by asymmetric pairings (cf. Section 3.2.1) which provides scalability to schemes with higher security requirements. In their paper, Chen et al. [16] made some security corrections (cf. [154]) over the previous proposal and propose a new asymmetric pairing-based DAA protocol together with a highly detailed security proof. This new scheme allows a much more efficient signature implementation in terms of computational complexity.

Recently, the DAA schemes based on SDH assumption are proposed which are more efficient than the previously developed schemes. In SDH-based proposals, each TPM chooses a unique membership key f, known only to TPM itself, and obtain a credential

[19] abbreviation BCL is used for bilinear Camenisch-Lysyanskaya signature scheme

on this key from the issuer. This credential is a SDH triple (A, x, y), which is considered as BBS+ [155] signature on secret key f. In order to provide a DAA signature, Host-TPM pair generates a signature proof of knowledge of such a SDH-triple together with a revocation token K used to check if the member in question is revoked.

Such a signature can be obtained similar to group signature counterpart as follows; First, an admissible bilinear pairing is selected as $e : G_1 \times G_2 \rightarrow G_M$. Then, credential issuer selects its private key $\gamma \in Z_q$ and compute its corresponding public key $w = \gamma P_2$ along with other public parameters, $P_1, H_1, H_2 \in G_1$ and $P_2 \in G_2$ where q is a large prime number. Although in schemes [65, 148] membership certificates are computed as a BBS+ signature such that $(\gamma + x)A = P_1 + fH_1 + yH_2$ holds, in [67] it is proven that SDH credential pair (A, x) computed as $(\gamma + x)A = P_1 + fH_1$ satisfies the necessary security requirements. In addition to the signature proof of knowledge of such a certificate obtained for the secret key, TPM generates revocation token K[20], computed as $K = fJ$ where J is computed from a basename or randomly from group G_1[21].

Revocation of existing members in DAA schemes are overlooked in previous proposals, only recently the issue of revoking illegitimate members has been extensively addressed by Chen and Li [156].

[20] Which is also used to provide user-controlled linkability

[21] Or another cyclic group selected for that purpose

Chapter 5

A^2-MAKE: Anonymous and Accountable Authentication Framework for Wireless Mesh Networks

In this chapter, we will describe in detail a framework named A^2-MAKE, which achieves seemingly conflicting privacy/security/anonymity and accountability goals at the same time. Although the framework is designed for wireless mesh networks (WMNs), it may also be applied to other wireless adhoc networks.

In Section 5.1, introduction and motivation behind the framework proposed for WMNs is given and the related work on WMN related privacy solutions are surveyed. In Sections 5.2 and 5.3, our construction is introduced starting with the explanation of the network architecture and problem formulation. Then, detailed description of our security framework for privacy preserving authentication and key establishment is given. User accountability provided via identification and revocation procedures is introduced in Section 5.4. In Section 5.5, security and privacy properties along with the performance analysis of the scheme are examined. In Section 5.6, implementation of the framework together with the corresponding timing analysis is discussed. The last section, Section 5.7, analyzes the simulation results of the introduced framework.

5.1 Introduction

Multi-hop hybrid wireless mesh networks (WMNs) have recently attracted increasing attention. For easy acceptance and wide deployment of WMNs, security, privacy, and accountability issues have to be addressed by providing efficient, reliable, and scalable

protocols. The fact that regular users, which may be resource-constrained wireless devices, are involved in routing activities highlights the need for efficiency and compactness. However, the objectives, security, privacy, accountability, efficiency etc., are, most of the time, not compatible. So far no previous work has adequately reconciled these conflicting objectives in a practical framework.

In the following, we present the design features and implementation of a framework named A^2-MAKE, which is a collection of protocols. The framework provides an anonymous mutual authentication protocol whereby legitimate users can connect to network from anywhere without being identified or tracked unwillingly. No single party (or authority, network operator, etc.) can violate the privacy of a user, which is provided in the given framework in the strongest sense. Our framework utilizes group signatures, where the private keys and corresponding credentials of the users are generated in a secure three-party protocol. User accountability is implemented via user identification and revocation protocols that can be executed by two semi-trusted authorities, one of which is the network operator. The assumptions about the trust level of the network operator are relaxed with respect to similar protocols. Our framework makes use of more efficient signature generation and verification algorithms[1] in terms of computational complexity than their counterparts in literature, where signature size is almost the same as the shortest signatures proposed for similar purposes so far.

5.1.1 Introduction and Motivation

In order to ensure wide user-acceptance and deployment of WMNs, *security* and *privacy* concerns of users need to be addressed in an efficient and reliable manner. Effective access control mechanisms that guarantee the registered users a reliable network connectivity and other security services for the protection of network communication are essential due to the dynamic and open nature of the network. Nevertheless, the services delivered to users may violate their privacy as they need to be authenticated to connect to the network. Another related issue is user *accountability* which aims to detect misbehaving users and,

[1] A variation of a direct anonymous attestation scheme [16] is utilized where both signature generation and verification operations are computationally efficient.

if needed, deny network access and other services via revoking. However, access control, security, user privacy and accountability may be conflicting objectives which are difficult to reconcile within the same framework.

The following real-world example due to Ren and Lou [4] highlights the need for a security and privacy aware framework in WMNs;

> '...at Boston suburb area, the City of Malden, the police department will use the WMN "to stream video footage from local areas directly to the police station, making it easier for police officers to monitor and respond to crimes at those locations" [157]. Obviously, all these communications contain various kinds of sensitive user information like personal identities, activities, location information, financial information, transaction profiles, social/business connections, and so on. Once disclosed to the attackers, these information could compromise any user's privacy, and when further correlated together, can cause even more devastating consequences....'

Therefore, in WMNs, it is essential to provide legitimate, privacy-aware network users with *anonymous* access to the network and other related services while unauthorized access must be prevented. It is not immediately obvious as to how to block unregistered users when everybody is anonymous in the network. Furthermore, protecting the network against misbehaving users requires *identification* capability built into network to achieve user accountability, whereby users are held accountable for their (unacceptable) actions. Identification capability and anonymity are, indeed, conflicting goals since, while the latter is trying to hide the user identity, the former is trying to reveal it.

In this chapter, we introduce how A^2MAKE manages these conflicting objectives successfully. More formally, the following security and privacy requirements are the objectives efficiently achieved in our framework;

- ***Confidentiality***: The framework incorporates an efficient key establishment protocol for protecting communications between a user and connecting router (or relaying user).

- ***Authentication***: Legitimate users anonymously authenticate themselves to connect to the network.

- ***User Privacy***: For user privacy, there are two requirements that need to be satisfied; anonymity and unlinkability. User-controlled linkability is actually provided as an optional requirement.

- ***User Accountability and Revocation***: Users should be held accountable for their malicious activities and should be revoked and prevented from connecting to a network and accessing the services provided. In our framework, we implement the opener, an entity to identify and revoke such malicious users, using two non-colluding semi-trusted parties, namely network operator and semi-trusted third party. The opening capability is distributed in order to avoid a fully trusted single *opener*. We postpone the discussion as to how this trust is implemented and managed to subsequent sections. The revocation protocol is applied to users whose subscriptions expire or who are accused of acting maliciously while the backward security and privacy is provided for all revoked users.

Our framework is practical and its protocols outperform previous protocols proposed for WMNs in literature [4]. Implementation and network simulation results of the protocols clearly demonstrate the feasibility and practicality of the framework.

5.1.2 Related Work

A related framework for an accountable and anonymous authentication is proposed by Tsang et al. [158], in which service providers (SPs) authenticate users. In that framework, there is no trusted third party (TTP) and accountability is provided by checking a blacklist held at SP side. Thus, the framework provides accountability on the SP side only. Therefore, it is not suitable for WMNs, where distributed accountability is required. Besides, although the scheme may well be adopted to WMNs, the signature size is more than twice of the signature size of the scheme proposed in A^2-MAKE and communication complexity depends on the number of blacklisted users by authenticating SP. Since

communication consumes much more energy than computation, it is desired to have the total size of the communicated values to be as small as possible.

Ren and Lou [4] proposed a closely related framework, which is one of the earliest studies on a privacy-enhanced authentication and key agreement scheme for wireless mesh networks. The framework is called PEACE; an abbreviation for SoPhisticated privacy-Enhanced yet Accountable seCurity framEwork for WMNs. PEACE is the first scheme to demonstrate that two conflicting goals, namely user privacy and accountability, can co-exist in a practical and efficient framework. In PEACE, privacy providing authentication is achieved through the use of short group signature scheme introduced in [58].

In PEACE, the network consists of a Network Operator (NO), a Trusted Third Party (TTP), a set of Group Managers (GMs), a set of mesh routers (MRs) deployed by NO, and a set of Network Users (NUs). Users are arranged in groups where there is one group manager for each group. User private keys (primarily for user authentication) are generated by the network operator and separate parts of the keys are given in a secure manner to the TTP and the corresponding GM. Neither the GM nor the TTP can fully recover users' private keys alone. A group manager assigns those keys to network users in its group via a protocol known as *late binding*. Then, each user reconstructs her private key by obtaining its shares from the TTP and her GM. Thus, although NO knows all the keys and private key-group manager mappings, it has no knowledge regarding to whom the GM assigns those keys. As a result, NO can trace a signature only up to the group of the user but not the specific user of a given signature.

In PEACE, group manager GM_i of group i, initiates a protocol with the NO to generate n private keys for users of the group i, where n is the number of users registered in that group. These keys are used in user-user and user-router anonymous authentication protocols before user gets access to the network. In this respect, NO generates n private keys and splits each key into two mathematically related shares and sends one part of the private key to GM_i and the other part to the TTP. Neither TTP nor GM_i alone can reconstruct the user private keys without knowing the private key of the NO.

Privacy against the NO is achieved via *late binding* of private keys by group managers to their corresponding users. Simply put, in late binding the group manager determines

which user will get which private key, and with the help of the TTP, a user in the group will be able to reconstruct her designated key. The NO is not involved in late binding process, and thus does not know which user gets to possess which private key. The NO is able to extract a private key used in a group signature produced by a user, and determine the corresponding group to which the user has registered. Nonetheless, it cannot trace it to the specific user who actually generates the signature as a result of the late binding process. However, if any two of the three parties, i.e. NO, TTP, and GM, collaborate, privacy can be compromised for any given user.

Although the NO cannot reveal the identity of a specific user by only knowing the key used in a signature, it can trace any signature up to its group and use this information to violate the anonymity of the signer. Furthermore, the NO can link two anonymous signatures if they are generated by the same user, and thus track down users without actually knowing their identity. The question here is ”Is it sufficient to hide the identity of the user to protect his privacy?” This issue is reminiscent of the infamous AOL Internet web search data release case. Privacy breach in AOL case is mentioned in [159] as;

> ’...search by search, click by click, the identity of AOL user No:4417749 became easier to discern...It did not take much investigation to follow that data trail to Thelma Arnold...’

In this incident, an AOL user whose identity was suppressed was easily tracked down and identified through the web pages she visited. In summary, if we de-identify a user but allow her to be tracked, then we violate the privacy of that user. From this point of view, PEACE allows the NO to track down the users in the network. Since NO deploys the access points and mesh routers and forms a well-connected wireless mesh network, it can collect valuable data such as location and time of users’ connections to the network. Moreover, NO does not have to search all the private keys, since it can immediately tell the group that a user belongs to. All NO has to do is a search within that group.

Conclusion, then, is that user private keys should not be given to or generated by a single entity, especially the network operator due to its advantageously situated position (i.e. it deploys the access points and routers thus establishing the whole WMN). Furthermore,

the NO, generally is not the best choice for acting as the authorized party that we can easily bestow the trust of users, and one must consider the requirements and cost associated with bearing such trust. Naturally, there are other techniques such as blind signatures that allow user private keys to be chosen and known only by users themselves. However, user accountability cannot be provided in such schemes.

One last comment on PEACE is that the verification algorithm adopted by the scheme needs to check whether the signer is in user revocation list (UserRL) by computing two pairings per user in the list[2]. This degrades the performance of the verification algorithm rendering the operation impractical for networks with large number of users. Therefore, a more efficient user revocation list checking algorithm is needed to enhance the performance of the security framework.

5.2 Network Architecture and Problem Formulation

In this section, we describe the network architecture, and then give corresponding construction details in the subsequent section. Our WMN architecture comprises four entities; a network operator (NO), a third party (TP), a set of mesh routers (MRs), and a set of Network Users/Mesh Clients (NUs).

In our framework, NO and TP[3] are assumed to be semi-honest parties [160]. Network Operator is semi-honest in the sense that it follows the rules of the protocol steps, but can launch an attack on the privacy and security of the user by recording any value it generates and/or receives during the protocol. Similarly, STTP is a semi-honest party in the sense that it also follows the rules of the protocol steps, but can record the values it calculates, generates and/or receives in the course of performing the protocol. In addition, it does not invoke the identification and revocation protocols on its own in order to violate user privacy.

Similar to PEACE, the NO deploys a number of access points and mesh routers in

[2]Efficiency of revocation checking can be improved by the modification mentioned in [58] but one must relinquish from some aspects of anonymity which is the utmost important requirement for the proposed authentication scheme

[3]This entity is referred as a Semi-Trusted Third Party (STTP) hereafter.

order to provide network services to users. Network users subscribe to NO to use the network from anywhere within the WMN. In order to provide network access only to legitimate users, and to protect network against malicious users, the NO must authenticate them via mesh routers. In addition, whenever it detects a misbehaving user or whenever the user's subscription period ends, the NO revokes that user and denies her further access to the WMN. Naturally, the NO cannot be trusted to perform the revocation process by itself since this, most of the time, means the compromise of the user identity. Therefore, we stipulate that a revocation process requires involvement of a STTP besides the NO.

In hybrid WMNs, users connect to the network through not only mesh routers, but also through other users already connected to the network. Users that act as routers should also be able to authenticate the other users that are outside the range of mesh routers, but still need to connect to the network. In addition, users must use necessary cryptographic means to protect their communication against eavesdropping, altering and sophisticated attacks aimed to compromise their privacy. As a result, there is a need for a privacy preserving mutual authentication scheme with revocation capabilities for anonymous and accountable authorization of users and for a key agreement scheme to provide confidentiality and integrity for the sensitive information exchanged within the network. Since the authentication operation is mainly based on signature schemes and is needed to be frequently performed, both signature generation and verification procedures must be efficient.

Similar to group signatures, users are issued private keys and associated credentials for anonymous authentication. Users can authenticate themselves by delivering a proof of knowledge for their private keys and the associated credentials. In our scheme, we employ a STTP to play the role of the issuer. The primary job of the STTP is to perform system setup and then to participate in the Join protocol (cf. Section 5.3.2) to generate user private keys along with corresponding credentials. Naturally, it needs to be involved in user identification and revocation operations as well.

In order to provide confidentiality and integrity, a key agreement step is incorporated into authentication scheme to reduce the communication and computation complexities.

Since the NO has full control over the WMN, it can track all the communications of

its own interest. Thus, in case of the NO being the credential issuer, it must provide this credential to a user anonymously. Therefore, during the Join protocol the user commits to a private value which is known only to herself. However, this means that the NO has no revocation capability when it is needed for accountability purposes (nor does anyone else). In other words, revocation and consequently user accountability are non-existent in the network. The PEACE protocol solves this problem by late binding, where the NO and the TTP (or group manager) collaborates to revoke a user. However, the fact that the NO knows all the private keys fully, and therefore can track users, may not be acceptable in applications where users are conscious of their privacy.

A^2-MAKE addresses these problems utilizing a three-party Join protocol involving the NO, the STTP, and the user herself. This protocol gives the user her private key securely while the NO and the STTP obtains a share of it (without knowing anything about the other party's share). The credential for the private key is another product of the Join protocol that is sent to the user.

Users that have the private key and the associated credential can perform two-party mutual authentication and key agreement protocol with optional linkability[4] by mesh routers and other users. User accountability is achieved through user identification and revocation. To this end, we provide two protocols: i) one that identifies the user and ii) the other that revokes the user private key, therefore the user herself. Naturally, both NO and STTP should give consent to and participate in identification and revocation operations.

We have two important assumptions on STTP and NO: They do not collude and both are semi-trusted parties that follow the steps of the protocols. This is a relaxation compared to fully trusted model where trusted parties are usually in possession of private keys as is the case with [4]. An entity which is similar to a certificate authority (CA) in classical public key setting is an example as to how STTP is implemented in real world. Since user registration is performed once for every user and revocation of users is needed occasionally, STTP does not have to be highly accessible.

[4]Part of the framework named MAKE

5.3 Our Construction

In this section, we give a detailed explanation of our anonymous and accountable mutual authentication and key agreement framework, A^2-MAKE, that consists of five protocols (*Setup, Join, MAKE, Identify, and Revoke*). Our mutual authentication protocol, MAKE, employs the algorithms used in DAA signature scheme proposed by Chen et al. [16]. This scheme incorporates Camenisch and Lysyanskaya signature scheme [60] adapted to asymmetric pairing setting and introduces/uses blind bilinear LRSW assumption, which is basically the blinded version of the bilinear LRSW assumption (cf. Definitions 23 & 22).

In the following, we specify the detailed steps of the first three protocols of A^2-MAKE. The last two protocols are explained in another section.

5.3.1 Setup

Given the security parameter 1^k as input, STTP performs the following steps:

1. Generates two additive groups G_1, G_2 of prime order $q \approx 2^k$ for which an asymmetric pairing is defined. The integer k is selected in such a way that solving decisional Diffie-Hellman problem (DDHP) and Gap-DLP [16] in G_1 is computationally hard,
2. Selects two generators P_1, P_2 of G_1, G_2, respectively; i.e. $G_1 = \langle P_1 \rangle$, $G_2 = \langle P_2 \rangle$,
3. Selects a pairing such that $\hat{e} : G_1 \times G_2 \mapsto G_M$, where G_M is a multiplicative group of order q and the DLP in G_M is computationally hard,
4. Determines hash functions $\mathcal{H}_1 : \{0,1\}^* \mapsto \mathbb{Z}_q$ and $\mathcal{H}_2 : \{0,1\}^* \mapsto G_1$ along with a key generating function $\mathcal{H}_K : G_1 \mapsto \mathbb{Z}_q$,
5. Generates its own public and private keys as follows;
   (a) Selects two random integers, $x, y \in_R \mathbb{Z}_q$ and sets them as its private key, namely (x, y),
   (b) Computes its public key: $(X, Y) \leftarrow (xP_2, yP_2) \in G_2$

6. Publishes public parameters, $\{G_1, G_2, G_M, \hat{e}, q, P_1, P_2, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_K, (X, Y)\}$

5.3.2 Join Protocol

Join protocol is used to provide a network user with a private key and an associated credential generated by the STTP, once the system parameters are set. The user can anonymously connect to the network using this private key and corresponding credential. The protocol is illustrated in Figure 5.1.

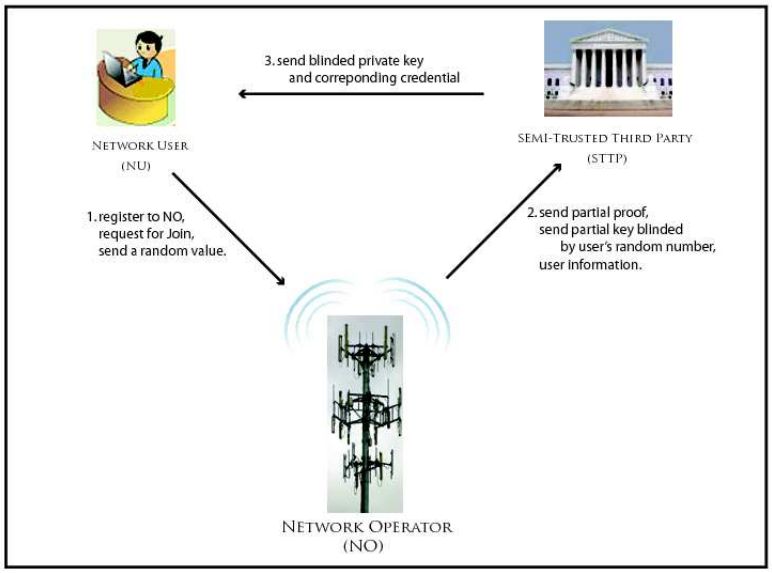

Figure 5.1: Join Protocol: Generation of Group Secret Keys and Associated Credentials

The protocol is a three-party protocol that involves the user, the STTP and the NO. The NO and the STTP jointly generate the user private key, which is fully known only to the user at the end of the protocol. The NO and the STTP keep random additive shares of the user's private key, which contain no information about the private key itself. They store these shares along with corresponding users' identities for future identification or revocation purposes, since the STTP and the NO need to collude to execute either identification or revocation operations. User's privacy is guaranteed against the STTP

and the NO, and henceforth she can anonymously authenticate that she is a legitimate member of the network.

In the following, protocol steps of the Join protocol for network user *i* (NU_i) are described in detail. Since the NO needs to know identifying information of the user during Join protocol in order to check if she is entitled to register for anonymous connection or not, conventional public key cryptography (PKC) is used for Steps 1, 2c, 3(b)iii, and 4a.

1. $\mathbf{NU_i}$ generates a random number $r_{US_i} \in_R Z_q^*$, encrypts and sends it to NO
2. **NO** (for user NU_i, where 'i' is the user identity)
   (a) Generates randomly its partial key share $f_{NO_i} \in_R Z_q^*$
   (b) Stores the mapping (i, f_{NO_i})
   (c) Encrypts and sends ($r_{US_i} + f_{NO_i}$) to STTP together with $f_{NO_i} \cdot P_1$
3. **STTP**
   (a) Generates the blinded key for NU_i
      i. Generates randomly its partial key share $f_{STTP_i} \in_R Z_q^*$
      ii. Stores the mapping (i, f_{STTP_i})
      iii. Calculates f_{temp}[5] $\leftarrow (r_{US_i} + f_{NO_i} + f_{STTP_i})$ and $f_{STTP_i} \cdot P_1$
      iv. Calculates F_{NU_i}[6] $\leftarrow (f_i \cdot P_1) \leftarrow (f_{NO_i} \cdot P_1 + f_{STTP_i} \cdot P_1)$
   (b) Generates the corresponding credential for NU_i
      i. Generates a random number $r \in_R \mathbb{Z}_q$
      ii. Calculates the credential
         – $A_i \leftarrow rP_1$, $B_i \leftarrow yA_i$, $C_i \leftarrow (xA_i + rxyF_{NU_i})$
         – $cred_i \leftarrow (A_i, B_i, C_i)$
      iii. Performs the encryption as $EC_i \leftarrow Enc_{PK_{NU_i}}(cred_i, f_{temp})$,
      iv. Sends EC_i to NU_i

[5]First decrypts the ciphertext received from NO to obtain ($r_{US_i} + f_{NO_i}$).
[6]This value is required in NU_i's credential generation process, 3(b)ii.

4. $\mathbf{NU_i}$

   (a) Decrypts EC_i and obtains $(cred_i, f_{temp})$, where $cred_i$ is her credential

   (b) Calculates her private key $f_i \leftarrow (f_{temp} - r_{US_i})$ and $E_i \leftarrow f_i \cdot B_i$

   (c) Checks whether the credential is generated appropriately:
   If $\hat{e}(A_i, Y) \neq \hat{e}(B_i, P_2)$ or $\hat{e}(A_i + E_i, X) \neq \hat{e}(C_i, P_2)$, then abort[7]. Otherwise, user can start using her private key and credential for subsequent anonymous authentication operations.

5.3.3 MAKE - Mutual Authentication and Key agrEement Protocol

MAKE allows a user to authenticate herself anonymously and gain access to the network and obtain a symmetric secret key to secure the link to the router once it is connected. It consists of three parts together with an optional step:

- **Key agrEement (KE):** User and router generate a mutual key using authenticated Diffie-Hellman key agreement procedure [5]. Note that the steps of the key agreement is incorporated into signature generation and verification steps.

- **Sign:** A user authenticates herself to connect to the network with an anonymous group signature. Upon receiving a beacon message[8] from a mesh router (or another user already connected in case there is no direct access to a router), user generates a signature that provides a proof of knowledge of her private key together with a corresponding valid credential for this key. The router also authenticates itself to the user with either anonymous group signature or conventional PKC.

- **Verify:** Router (or a relaying user) verifies the received signature from the connecting user. It first checks whether the user is in user revocation list (UserRL); and if the user is not in the list, then checks whether the signature verifies. If both checks are successful, it assists the user to connect to the network.

[7] This step is necessary also for checking the correctness of the private key, f_i.
[8] A specifically formed message indicating that the router (or the relaying user) is available.

- **Link (optional):** Linking phase is optional and whether to perform this phase or not is decided by the network user and relaying agent (either mesh router or another network user) together. If linking phase is performed, the user can be traced, which may be desirable to the user for a specific amount of time in some applications, e.g. user may want to continue a previous session. Linkability can also be utilized in systems such as privacy enhancing identity management [161] and to thwart attacks on networks providing anonymity [15]. Linking, which requires user's consent, is achieved by relating two signatures by the same user in MAKE protocol. Re-verification of the signatures may be needed to check the possibility of user being revoked after the time of the reception of the signatures.

In the following, we first analyze the case when a user tries to connect to the network using a mesh router, and then discuss the case, where another user acts as the router. For the latter case, user, which acts as a router in user-user authentication, should check UserRL to deny network access to revoked users. In the following, it is assumed that all agents acting as routers have access to the UserRL (have the means to obtain UserRL), and can perform UserRL check operation. UserRL may become a large list by time, so storing it and performing check operation on UserRL may be very expensive. An alternative solution in such cases is that the routing agent delegates UserRL check operations to another party who is more capable; e.g. having more storage, computation and communication resources, and higher connectivity.

- **MAKE for User-Router Interaction (MAKE-UR)**

  1. **MR** broadcasts a beacon periodically and an authentication payload is sent as part of this beacon (The following steps are almost the same as those in [4]):
     (a) Picks a random nonce $r_{MR} \in_R \mathbb{Z}_q$, a timestamp ts_{MR} and a random generator $P_{MR} \in_R G_1$
     (b) Chooses a basename $bsn_{MR} \in \{0,1\}^*$ to be used in providing optional user-controlled linkability
     (c) Computes $T_{MR} \leftarrow r_{MR} \cdot P_{MR}$

(d) Signs P_{MR}, T_{MR} and ts_{MR} using a conventional digital signature algorithm (e.g., ECDSA):
$\sigma_{MR} \leftarrow Sign_{SK_{MR}}(P_{MR},\ T_{MR},\ ts_{MR})$

(e) Broadcasts $\text{Msg}_{MR} \leftarrow \{P_{MR},\ T_{MR},\ ts_{MR},\ \sigma_{MR},\ Cert_{MR},\ bsn_{MR}\}$ as a part of the beacon

2. **NU**, upon receipt of Msg_{MR}, performs the following steps to authenticate MR:

(a) Checks if the timestamp ts_{MR} is fresh

(b) Validates the certificate of MR ($Cert_{MR}$) using Online Certificate Status Protocol (OCSP) [162, 163] or a similar protocol depending on the infrastructure[9].

(c) Verifies signature σ_{MR} generated by MR. If the signature is valid, then user accepts the router as authentic (non-anonymous authentication via conventional PKC).

3. **NU** authenticates to MR and initiates the authenticated key agreement algorithm:

(a) For symmetric key establishment,

i. Picks a random nonce[10] $r_{NU} \in_R \mathbb{Z}_q$ and computes $T_{NU} \leftarrow r_{NU} \cdot P_{MR}$

ii. Calculates the mutual key using key generating function $\mathcal{H}_K$:
$K_{UR} \leftarrow \mathcal{H}_K(r_{NU} \cdot \{r_{MR} \cdot P_{MR}\}) \leftarrow \mathcal{H}_K(r_{NU} \cdot T_{MR})$

(b) For signature generation,

i. Generates timestamp ts_{NU} to prove freshness

ii. If linkability is to be provided, then NU gets the router specific basename, bsn_{MR}, which is provided by the router within the beacon message and computes $J = \mathcal{H}_2(bsn_{MR})$. Otherwise it generates J as a random point, i.e. $J \in_R G_1$

[9]Note that this validation requires network user to be connected to the Internet. So, if the service provided by the mesh router is the Internet service, which is mostly the case, then in Join protocol, the (conventional) certificate revocation list (or a similar list) for the routers must be given to each user, or the list must be provided by the router in MAKE protocol which should have been signed by NO or STTP.

[10]All nonces used by Network User within the protocol are *randomly* generated in each session to prevent linkage of any kind.

iii. Generates a random number $t \in_R \mathbb{Z}_q$ to randomize the credential

iv. Randomizes the credential

$(A',\ B',\ C') \leftarrow (t \cdot A,\ t \cdot B,\ t \cdot C)$

v. Calculates signature proof of knowledge

- $K \leftarrow f_{NU} \cdot J$
- Selects a random value $z \in_R \mathbb{Z}_q$
- Calculates pairing value ρ_D to be supplied into the challenge c together with the witness value L:

  $\rho_D \leftarrow \hat{\mathrm{e}}(z \cdot B', X)$ and $L \leftarrow z \cdot J$
- Calculates the challenge value

  $c \leftarrow \mathcal{H}_1(params^{11}||A'||B'||C'||J||K||L||\rho_D||K_{UR}||ts_{MR}||ts_{NU})$
- Calculates the response value

  $s \leftarrow z + c \cdot f_{NU} \pmod q$

vi. Assembles the signature σ_{NU}

$\sigma_{NU} \leftarrow (A',\ B',\ C',\ K,\ J,\ c,\ s)$

vii. Sends signature σ_{NU} together with DH key agreement share, T_{NU}, and timestamp ts_{NU}

$\text{Msg}_{NU} \leftarrow \{\sigma_{NU},\ T_{NU},\ ts_{NU}\}$

4. **MR** verifies the user anonymously and obtains the shared key K_{UR}:

(a) Checks if the timestamp ts_{NU} is fresh

(b) If linkability is to be provided, then it checks whether the random point J is formed correctly, i.e. $J = \mathcal{H}_2(bsn_{MR})$

(c) Checks if NU is in UserRL

If $\exists f_i \in UserRL$, such that $K = f_i \cdot J$, then rejects the signature and aborts the protocol.

(d) Checks the correctness of A' and B':

If $\hat{\mathrm{e}}(A', Y) \neq \hat{\mathrm{e}}(B', P_2)$, then rejects the signature and aborts the protocol.

[11] Publicly known parameters, i.e. public keys of STTP, that are required to be included in challenge calculations

(e) Computes the shared secret key

$K_{UR} \leftarrow \mathcal{H}_K(r_{MR} \cdot \{r_{NU} \cdot P_{MR}\}) \leftarrow \mathcal{H}_K(r_{MR} \cdot T_{NU})$

(f) Verifies the Signature (Correctness of Proofs)

i. Performs the following computations

$\rho_A^{'} \leftarrow \hat{\mathrm{e}}(A', X)$, $\rho_B^{'} \leftarrow \hat{\mathrm{e}}(B', X)$, $\rho_C^{'} \leftarrow \hat{\mathrm{e}}(C', P_2)$

$\rho_D^{'} \leftarrow (\rho_B^{'})^s \cdot (\rho_C^{'}/\rho_A^{'})^{-c}$, $L' \leftarrow sJ - cK$

ii. Validates the challenge

If $c \neq \mathcal{H}_1(\text{params}||A'||B'||C'||J||K||L'||\rho_D^{'}||K_{UR}||ts_{MR}||ts_{NU})$, then rejects the signature and aborts the protocol.

(g) Assists the user to connect to the network.

5. **MR**, if user-controlled linkability is opted, determines whether a given pair of signatures are generated by the same user (whether they are linked or not[12]).

Given a pair of signatures, σ_0 and σ_1;

(a) Verifies signatures, σ_0 and σ_1. If any one of them is rejected, then algorithm returns that the signatures are *unlinked*.

(b) Compares the corresponding J and K values. If they are matched, $J_0 = J_1$ and $K_0 = K_1$, then algorithm concludes that the two signatures are generated by the same user (i.e. they are *linked*). Otherwise, if any one of the two equations is not satisfied, algorithm returns that the signatures are *unlinked*

Note that the given signatures are verified in linking step, even if they are proved to be valid previously by the verifier. This is so, due to the possibility of the revocation of the network user in consideration who is previously considered legitimate.

Upon successful completion of the protocol, user and router can use the shared secret key K_{UR} to secure further communication between them.

[12]This step is only performed if linkability is to be provided and can either be performed within the MAKE protocol or treated as a standalone step.

- **MAKE for User-User Interaction (MAKE-UU)**

  In case a user cannot find a router within its reception range, but finds another user already connected to the network, the two users can run a similar protocol. The only difference from the previous scheme (i.e. MAKE-UR) is that the relaying user also provides an anonymous group signature in his beacon to authenticate herself. Already connected to the network, the user has a private key and corresponding credential for anonymous authentication. As a result, both users mutually authenticate each other anonymously using their private keys along with the related credentials they acquire in Join protocol[13].

5.4 User Accountability and Key Revocation

User accountability is possible through two important capabilities that are incorporated into the framework: identifying and revoking users. Below, we discuss what they exactly mean and how they are implemented.

- **User Identification** For user accountability, it is necessary to identify misbehaving users. In this respect, our first proposed protocol, *Identify*, is designed so that the NO and the STTP can reveal the owner of a given signature, only if they collude. *Identify* is a two-party protocol, whereby the STTP extracts the identity of the user who is the owner of the given signature(s) without obtaining the user's private key. This is a useful property since the user still remains anonymous to the NO and can continue connecting to the network. In addition, signatures from this user also remain anonymous to both NO and STTP since the user's private key is not extracted by running this protocol. The NO can use this protocol only when there is a suspicious activity or a dispute. It is up to the STTP to hand over the identity of the signer to any other party. Besides, since user's private key is not revealed, she cannot be revoked and there is no need to re-execute the Join protocol for the user if the case is not pursued any further.

[13] In this case, it is assumed that all users have access to the UserRL in order to perform verification of the anonymous signatures provided by both sides.

- **User Revocation** This is a protocol basically built upon the Identify protocol, whereby on a given signature, the STTP and the NO identify and revoke the user by adding her private key to user revocation list (UserRL). In this protocol, user is identified and corresponding key is extracted by the STTP, which has the authority of revoking the user. During authentication, the UserRL is checked to make sure that the signer is not in the list. UserRL is only updated and signed by STTP and privacy of other users remains unaffected by the revocation process. User revocation can be applied also to users whose subscription to the network expires. UserRL does not contain user's real identity but her private key only, and therefore a user can get another private key after her subscription ends.

For user accountability, situations may arise, where it is required to identify a user suspected of possible malicious behavior. If the user is actually found guilty for malicious activity, then it becomes imperative to add its private key, f_i for user NU_i, to the UserRL; in order to revoke the user NU_i. UserRL contains only the private key of a user, and not her identity. Therefore, this private key cannot be used anymore, but the user may acquire a new key if she proves her innocence after revocation has occurred. Besides, a user whose subscription ends can get a new private key when her subscription is renewed by re-performing the Join protocol.

In addition, circumstances may also occur, where legitimate users' keys are compromised by attackers. In those circumstances, compromised users can initiate the key revocation protocol by revealing their private keys to STTP, which adds these compromised keys to the UserRL. For this, STTP can also collaborate with NO to perform the revocation operation without users supplying their private keys.

Note that both STTP and NO maintain a list of pairs, (i, f_{STTP_i}) and (i, f_{NO_i}), respectively. The integer value, i, is used by both parties to refer to a user, and need not be related to its real life identity. Below, the detailed steps of the mentioned protocols are explained.

5.4.1 Identify - (User identification without private key extraction)

In this protocol, STTP, collaborating with NO, identifies the signer of a given signature without actually extracting her private key. The STTP needs a user signature, σ_O, to identify its owner. For this purpose, STTP and NO perform the following protocol steps;

1. **STTP:**

   (a) Verifies the signature σ_O, where $\sigma_O = \{A_O, B_O, C_O, J_O, K_O, c_O, s_O\}$

   (b) If the signature verifies, then it sends J_O to the NO and requests for the corresponding partial proofs (i.e., $f_{NO_i} \cdot J_O$ for all registered users).

2. **NO**, upon receiving J_O:

   (a) Calculates partial proofs for every registered user $NU_i \in RU$, where RU stands for the list of registered users and $|RU| = n$
   $\{\forall\ NU_i \in RU,\ f_{NO_i};\ K_{NO_i} \leftarrow f_{NO_i} \cdot J_O\}$

   (b) Sends n proof pairs (i, K_{NO_i}) to STTP.

3. **STTP**, using the proof pairs received from network operator:

   (a) Calculates corresponding partial proofs using secret shares in its own list:
   $\{\forall\ NU_i \in RU,\ f_{STTP_i};\ K_{STTP_i} \leftarrow f_{STTP_i} \cdot J_O\}$

   (b) Calculates proofs by adding partial proofs K_{STTP_i} and K_{NO_i} and compare the result with $K_O (= f_i \cdot J_O)$:

      i. $\forall\ NU_i \in RU$, calculate $K_i = K_{STTP_i} + K_{NO_i}$ and check if $K_i = K_O$

      ii. If $\exists i$ for which $K_i = K_O$ then output *i* as the corresponding signer

STTP outputs the user id *'i'* only if it is necessary and otherwise keeps it secret and discards the signature and all related values.

5.4.2 Revoke - (User revocation with private key extraction)

If it is decided to revoke the signer of a given signature σ_O, then signer's private key is uncovered by STTP and NO together. In order to perform this task they perform the following protocol steps;

1. **STTP** initiates *User Identification* protocol using σ_O and gets user identity *'i'*
2. **STTP** asks for NU_i's partial private key value from NO by sending user id *'i'* to NO.
3. **NO** sends corresponding private key share f_{NO_i} to STTP
4. **STTP**, upon receiving the partial secret:
   (a) Computes the private key $f_i \leftarrow (f_{STTP_i} + f_{NO_i})$ of NU_i.
   (b) Adds f_i to UserRL and corresponding user id *'i'* to another list in case where STTP wants to prevent the revoked user from re-performing Join protocol in the future.

5.5 Security and Performance Analysis

In this section, we give security and performance analysis of our mutual authentication and key agreement architecture. The proposed architecture provides user-router mutual authentication where the user remains anonymous after the authentication, and user-user authentication whereby both ends of the communication remain anonymous after the authentication.

5.5.1 Security Analysis

In our construction, we assume that there exist pairwise secure channels connecting the NO, the STTP and the user during the Join protocol where all exchanged information is protected. Since privacy and anonymity is not an aim in Join protocol, its security can be provided using conventional cryptographic methods.

In the following, security properties provided by A^2-MAKE are explained.

- ***User anonymity against other users, NO, and STTP***

  Our construction makes use of a variant of protocols given in direct anonymous attestation scheme of Chen et al. [16] to protect the anonymity of a user against the other users, mesh routers, the network operator, and even against the STTP. Since no single entity within the network knows the private key of any user but the user herself, no one is able to identify the owner of a given signature or link signatures generated by the same user. The STTP cannot link two signatures by the same user (even if STTP records the credential-user pairs) since the credential of a user is re-randomized in every authentication session. To identify, track (by linking signatures) and revoke a user, the NO and the STTP have to collaborate to run identification and revocation protocols successfully.

- ***Confidentiality and Integrity***

  Communicating entities establish a shared symmetric secret key to secure their communications. In our proposal, we use authenticated Diffie-Hellman key exchange procedure to establish such a symmetric key between the communicating parties. A user that wants to connect to the network should always generate random nonces to make sure that a different secret key is generated in every session. The secret key derived in our scheme secures only the communication channel between the user and the router.

- ***User Accountability***

  User accountability is made possible by the revocation capability incorporated into the scheme. Whenever a malicious activity is observed, it can be reported to the STTP via providing a signature used by the malicious user for authentication. STTP and NO need to collude to recover the identity of the owner of the signature. Then, in accordance with the situation, STTP decides on whether to revoke reported user's secret key or not. In addition, NO can easily invalidate user subscription by utilizing the revocation protocol.

In the following, we discuss the security details of our protocols and their steps;

1. **Join Protocol :** The Join protocol utilizes a *Secret Sharing* scheme in which private key (f_i) of a network user is jointly constructed by and the secret shared between NO and STTP. NO generates its share of user's private key randomly, f_{NO_i}. It then blinds that partial key with the random number (r_{US_i}) received from the NU_i, which is referred to as *blinding key*. Since NU_i sends its blinding key encrypted by the public key of NO, nobody other than the NO can see the blinding key. NO encrypts and sends its blinded share ($r_{US_i} + f_{NO_i}$) to STTP. After decrypting the received message, STTP adds its own random share, f_{STTP_i} to ($r_{US_i} + f_{NO_i}$). It then sends $f_{temp} = (r_{US_i} + f_{NO_i} + f_{STTP_i})$ to NU_i, which is the only person that can extract the private key $f_i = f_{NO_i} + f_{STTP_i}$. This scheme is secure under two assumptions: i) NO and STTP are *semi-honest* parties in the sense they follow the protocol steps and ii) they are *non-colluding*, which means here that they do not betray their secret shares of user's private key to each other. These assumptions are common in many cryptographic protocols [160].

   NO, along with $(r_{US_i} + f_{NO_i})$, sends $(f_{NO_i} \cdot P_1)$ to STTP. This is secure under ECDLP assumption, since STTP is required to solve the elliptic curve discrete logarithm problem in G_1 to get the NO's share, f_{NO_i}, from the value it receives. ECDLP assumption implies that solving DLP in G_1 is computationally hard.

   As can be seen in Step 3(a)iv of Join Protocol, by using the corresponding point share $(f_{NO} \cdot P_1)$ of the NO, STTP can compute the blinded key $F_{NU_i} = (f_i \cdot P_1)$ of NU_i, which is needed to generate the credential. Private key of NU_i is protected against STTP by the same DLP assumption in G_1. In Step 3(b)iv of Join protocol, STTP sends the ciphertext EC_i to NU_i, which contains f_{temp}. If NO can capture and decrypt EC_i, it can compute NU_i's private key f_i since it knows the blinding key r_{US_i}. However, NO cannot decrypt EC_i without knowing the private decryption key of NU_i, and therefore f_i is protected against disclosure by NO via encryption.

   Lastly, since network user checks the correctness of the private key, neither STTP nor NO is able to manipulate the random private key generation process and they

are obliged to use the random number provided by the network user for blinding purposes (see Step 4c and related footnote).

In summary, security of the Join protocol relies on the following assumptions:

- NO and STTP are semi-honest and non-colluding,
- Underlying encryption scheme is assumed to be secure against adaptive chosen ciphertext attack in the random oracle model,
- ECDLP assumption in G_1.

2. **MAKE :** An active router broadcasts *beacon* messages to indicate its availability to users who want to connect to the network. Here, we assume that the router is the mesh router (MR) deployed by NO in our security analysis. Msg_{MR} in the beacon is signed by MR using a conventional PKC digital signature algorithm. A user who wants to connect to the network through a MR checks the authenticity of Msg_{MR} by verifying the signature σ_{MR}, provided by MR within the message. To impersonate a legitimate MR, the attacker has to forge a valid signature on the message derived by the attacker itself. However, attacker is not be able to succeed under the UF-CMA[14] [8] security assumption of the underlying digital signature scheme utilized within the protocol.

   In order to secure the communication link between NU and MR, parties perform Key agrEement (KE-part of MAKE) protocol to generate a shared key. To do this, MR generates a random number, r_{MR}, which is the contribution of MR to the mutual encryption key K_{UR}. Then, it includes the elliptic curve point $T_{MR} = (r_{MR} \cdot P_{MR})$ in the beacon, used in the Diffie-Hellman key agreement protocol. In her response, user also sends its own share to MR during the protocol. So, security of the KE-part of our protocol is guaranteed under the hardness of computational Diffie-Hellman (CDH) problem (cf. Definition 16). Therefore, for an attacker to be able to compromise the shared key and the communication between the two parties, it must solve elliptic curve CDH problem, which is believed to be hard.

[14] UnForgeable against Chosen Message Attack

Mutual authentication (MA-part of MAKE) in MAKE scheme is secure in the random oracle model if the following security assumptions hold [15]:

(a) **Blind Bilinear LRSW Assumption in ($\mathbf{G_1}, \mathbf{G_2}, \mathbf{P_1}, \mathbf{P_2}, \mathbf{e}$)**: This ensures that a valid (randomized) credential obtained from a user could have been generated only by STTP.

(b) **Hardness of Gap-DLP in $\mathbf{G_1}$**: In Step 3(b)v of MAKE-UR protocol, user computes a witness, $L \leftarrow z \cdot J$, for a given point J; and consequently provides a proof of knowledge $K \leftarrow f_i \cdot J$, using her private key f_i. Even if an oracle outputs $K' \leftarrow f_i \cdot J'$ for a given J', it is still difficult to learn f_i due to Gap-DLP assumption. In addition to the proof and the witness, user also calculates a response $s \leftarrow z + c \cdot f_i$ to the challenge c using again the secret key f_i. The response cannot be produced without the knowledge of f_i. As a result, no one is able to produce a valid anonymous signature without possessing a legitimate private signing key.

(c) **Hardness of the decision Diffie-Hellman problem (DDH) in $\mathbf{G_1}$**: User randomizes her credential to hide her identity in every authentication operation, i.e. $(A',\ B',\ C') \leftarrow (t \cdot A,\ t \cdot B,\ t \cdot C)$ for a randomly chosen $t \in_R \mathbb{Z}_q$. Here, DDH assumption is important to hide the fact that these two credential sets are related. In symmetric setting, where DDH is easy in G_1, one must take additional precautions against linkability of the credentials by utilizing extra randomness (cf. [64]) since anyone could easily solve DDH by performing four pairing operations, and checking whether the following equations hold, $\hat{e}(A', B) = \hat{e}(A, B')$ and $\hat{e}(A', C) = \hat{e}(A, C')$[16]. Therefore, from the efficiency point of view, asymmetric setting is specifically preferred, where DDH is hard in G_1, since our framework necessitates the hardness of determining whether given two credential randomizations belong to the same credential or not.

[15] For the detailed discussion of the logic for the requirement of these assumptions, reader is referred to [16]

[16] In symmetric setting, we can immediately tell that $c \equiv ab \pmod q$ for $\{P_1, aP_1, bP_1, cP_1\}$, if $\hat{e}(aP_1, bP_1) = \hat{e}(P_1, cP_1)$.

In order to protect players from replay attacks, the symmetric key, K_{UR}, generated in key agreement steps is included as an additional random nonce. Therefore, it is the responsibility of the network user to generate different random numbers every time she connects to the network. Since we integrate our signature scheme with key agreement, instead of generating additional nonces in the challenge calculation, we utilize the shared key obtained in Key agrEement together with the timestamp values, which do not incur additional communication cost.

Our MAKE protocol is based on the DAA scheme presented in [16], which is proposed originally for trusted platform modules (TPM [164]). We now explain how our protocol differentiates from the one in [16]. In TPM setting, for user entity, there is a user computer (host)-TPM pair while in our framework we have only a network user and there is no need for such a separation. This allows us to combine steps of the scheme taken separately by the host and TPM into steps performed by a single entity, i.e. the network user. In our protocol, instead of doing exponentiation in the extension field ($t = \beta^z$), we replace it with a single elliptic curve multiplication ($z \cdot B'$). We also utilize the timestamp values and symmetric key obtained from the KE-part as substitutes for the random nonces required. Thus, our anonymous signature protocol also serves as an authentication step for the key shared between two communicating entities. As a result, security proof of our signature scheme can be reduced to the proofs provided by Chen et al. [16].

Verification step of our protocol is nearly the same as the verification algorithm presented in [16]. The main distinction of our protocol is the generation of the revoked (rogue in TPM terminology) user list (UserRL). In our protocol, malicious users are revealed by STTP (with the help of NO) and their private keys are added into UserRL. In DAA schemes, how a private key is revealed is not described. The access to the network by revoked users is prevented via the UserRL checking performed by the relaying agent, either the router or the relaying network user. In addition to this, similar to the signature generation part, instead of generating and communicating nonces to be used in replay attack prevention, we utilize timestamps

along with the symmetric key obtained as a result of the key agreement protocol. We refer the interested reader to Chen et al. [16] for the detailed and formal security proofs of the underlying signature generation and verification protocols.

3. **User Identification / Revocation :**

   In case of a dispute or suspicious activity, any mesh router and/or network user acting as a relaying agent may call for the identification and/or revocation of the owner of a signature. It is STTP that has the ability to revoke the user in question. With the help of NO, STTP can identify the owner of a given signature without the consent of the user, and/or even revoke the user by adding her private key to UserRL.

   In user identification protocol, STTP cannot learn the private key of the user. NO sends only the partial proofs for a given signature to STTP, i.e. the points, $K_{NO_i} = (f_{NO_i} \cdot J)$, computed in Step 2c, which are elliptic curve points. Given these points, K_{NO_i}, for all registered users, the private keys of the corresponding users are protected by the assumption that ECDLP is hard in G_1.

   When linkability option is not adopted, all the values used in the signature construction are chosen randomly (see Step 3b of MAKE-UR). The only way to link two signatures by the same person is via the elliptic curve point $J \in G_1$ that is a part of the signature σ_{NU}. Since two randomly chosen elliptic curve points cannot be related, users' privacy is preserved. The linkability can only be achieved if the user agrees to compute this point under a given basename (see Step 3(b)ii of MAKE-UR). Otherwise, no one can link two signatures unless one can compute elliptic curve discrete logarithm in G_1.

   After identifying a user from its signature, STTP has only the identity of the user i, but not her private key, which is also protected under the assumption of the hardness of the ECDLP in G_1. STTP has only a secret share of the user's private key, f_{STTP_i}, and needs NO's share f_{NO_i} to construct it. It is at STTP's discretion to run the revocation protocol and add the private key to the UserRL. In our framework, we assume that STTP is endowed with the trust to make reasonable and fair decisions

pertaining to user revocation. Here, we do not specify how STTP makes these decisions since they depend on the policies that are adopted and agreed by the participants of the network. But we can sketch a sample situation below, where a user is revoked. If an anonymous user is suspected of malicious behavior or any other potentially harmful activity, NO can ask STTP to identify the user in question by providing sufficient proof pertaining to the malicious activity. If the user is identified several times for similar misbehaviors, NO and STTP can decide to revoke the user. Independent of the adopted policy for user identification and revocation, our framework provides the technical infrastructure to perform these operations efficiently and discreetly.

Backward Security and Privacy[17]: When a user is revoked, it is important to analyze what happens to her past communications. If the security of a user's past communications cannot be compromised by an adversary that records the transcripts of all messages sent and received by the user, we say that the system provides backward secrecy. In our scheme, secret symmetric keys are obtained via authenticated Diffie-Hellman key agreement using randomly selected secret numbers. This key agreement operation does not utilize private keys of users, which are added to the UserRL after revocation. Therefore, our scheme provides backward security since a revoked private key does not reveal any information about the symmetric key, K_{UR}.

On the other hand, an adversary that records signatures, σ_{NU}, can compromise the privacy of users after their private keys are revoked. However, if the signatures used in authentication, σ_{NU}, are encrypted by the symmetric key K_{UR}, the backward privacy is guaranteed against the parties that do not know these keys. A router that knows a secret key, K_{UR}, can only learn that a corresponding user is revoked if it records all signatures it verified in the past. It can never identify a revoked user since users' identities are not added to the UserRL. The router must record not only all signatures but also all secret keys it used to secure connections of all users it helped connect to the network. Then, it needs to try all revoked private keys in UserRL and all secret keys just to tell whether the user is revoked. Storing all these keys and doing all these computations may not be feasible

[17] see [165] for a discussion on backward and forward security concepts.

for routers. In summary, with simple encryption of signatures, the backward privacy of users against third parties other than routers are fully protected while it is only partially compromised by the routers.

5.5.2 Performance Analysis

In this section, we analyze the computational and communication overheads of the proposed framework. Since Join protocol is normally performed only once for a user, it is not a performance bottleneck. Moreover, communication steps of the protocol is protected by conventional cryptography as privacy is not an objective in Join protocol. Therefore, we focus on the complexity of the mutual authentication and key agreement protocol (MAKE), which needs to be performed efficiently. We compare our results with those in PEACE [4], which is the most related work. We start with computational overhead, where complex cryptographic primitives dominate the CPU time spent on A^2-MAKE.

5.5.2.1 Computational Overhead

Table 5.1 lists the operations performed by a network user during signature generation and by the router during signature verification in A^2-MAKE protocol. In this table, P, G_1, G_M^2, G_1^2 stand for a pairing operation, an elliptic curve point multiplication in G_1, a multi-exponentiation in G_M, and two simultaneous elliptic curve point multiplications in G_1, respectively[18]. Table 5.1 lists the operation count of signature generation and verification operations for the framework PEACE [4] for comparison purpose. PEACE does not use a protocol similar to Join, so Table 5.1 lists the operations performed by each party in Join protocol only for the proposed framework.

As can be observed in Table 5.1, our signature algorithm requires half the number of pairing operations compared to the signature scheme employed in PEACE [4]. Since pairing is usually the most time-consuming operation, saving obtained in our signature scheme is of great importance. Furthermore, considering that the signature generation is the most frequent operation a user performs, our protocol is more suitable for users with

[18] Note that multi-exponentiation and simultaneous elliptic curve point multiplications can be performed faster than executing these operations separately [166].

Operation	*Party*	*Cost - A²MAKE [167]*	*Cost - PEACE [4]*
Join	STTP	$3P + (2 + \|\text{UserRL}\|)G_1 + 2G_1^2$	-
	NU	$4P + 3G_1 + 1Sign$	
	MR	$6G_1 + 1P$	
Sign	NU	$1P + 8G_1$	$2P + 8G_1$
Verify	MR	$5P + \|\text{UserRL}+1\|G_1 + G_M^2 + G_1^2$	$(3 + 2\|\text{UserRL}\|)P + 6G_1$

Table 5.1: Computational Overhead of A^2-MAKE and PEACE [4]

constrained resources. Note that the weakest point in a network as far as the resources are concerned is users. Therefore, it is natural to optimize the operations for network users. Our algorithm clearly favors resource constrained network users.

Signature verification operation is performed by the router (or relaying network user) that helps users connect to the network. Table 5.1 lists the complexities of both the proposed framework and PEACE [4] in terms of the aforementioned operations. One important factor in the verification process is to check UserRL to see if the user is in the list. This check dominates signature verification operation for even a UserRL of relatively small size. While the number of pairing operations is proportional to the size of UserRL in PEACE protocol, in our scheme the number of elliptic curve point multiplications is proportional to the size of UserRL[19]. In addition, the number of checks increases in a slower fashion in our protocol than in PEACE (compare the terms $|UserRL|G_1$ and $2|UserRL|P$). For each user in UserRL, our protocol requires single additional elliptic curve point multiplication while this number is two pairing operations per user in PEACE. For $|UserRL| \geq 2$, verification step of A^2-MAKE is carried out with less computational overhead than the one performed in PEACE. Efficient verification algorithm for anonymous signatures is a crucial requirement in hybrid mesh networks where regular users also perform verification of anonymous signatures while they act as routers. So, it is an open problem to devise a revocation mechanism such that it does not depend on the number of revoked users in UserRL list.

One important note about the type of pairing operations must be given here. PEACE can use symmetric pairings over supersingular curves, which are faster than their asym-

[19]pairing operation is usually several times slower than elliptic curve point multiplication [168, 169].

metric counterparts, which our scheme utilizes. However, this difference quickly diminishes at higher security levels. Moreover, speed difference between symmetric and asymmetric pairings is not as important as the number of pairing operations.

In summary, both our signature generation and verification algorithms are more efficient that their counterparts in PEACE as far as the computational complexity is concerned. In the next section, we analyze the communication overhead of both the proposed and PEACE protocols.

5.5.2.2 Communication Overhead

Since WMNs' clients are resource constraint entities, and also since message transmission and reception operations are very demanding operations in terms of resource and energy, communication overhead due to authentication protocol (appended to the original payload) should also be minimized. This is, to a great extent, related to the size of the signature and other agreement values used in authentication and key agreement protocols.

In Table 5.2, total communication overhead of our protocol is given for both 80-bit and 128-bit security levels (using 160-bit and 256-bit elliptic curves, respectively). In calculating the total number of bits exchanged over the wireless link between a mesh client and a mesh router, it is assumed that ECDSA algorithm[20] is used for router authentication, the size of the timestamp values are 32-bits and router's ID as well as the optional basename are 128-bits. Furthermore, certificate $Cert_{MR}$ for the conventional signature is assumed to be composed of 320-bits (512-bits for 128-bit security level) of a signature and a 128-bit ID.

For comparison purposes, we also provide signature lengths of our protocol and of PEACE in Table 5.3. Note that signature lengths are a dominant factor in communicational complexities of both protocols.

Since elliptic curve points can be represented by its x-coordinate and an additional 1-bit of information pertaining to its y-coordinate, we may take communication overhead

[20]Signature size of the ECDSA is $4t$ ($= 2q$ where q is the order of the elliptic curve group) where t is the security level measured in bits. Thus, signature sizes are assumed to be 320-bits and 512-bits for 80-bit and 128-bit security levels, respectively.

Security Level	*Communication Overhead*	*Total Size*
MR-to-NU	$P_{MR}, T_{MR}, ts_{MR}, \sigma_{MR}, Cert, bsn^*$	$6q + 418$
80-bit	($q = 160$-bits)	$= 1378$ bits (≈ 173 Bytes)
128-bit	($q = 256$-bits)	$= 1954$ bits (≈ 245 Bytes)
NU-to-MR	$A', B', C', K, J, c, s, T_{NU}, ts_{NU}$	$8q + 38$
80-bit	($q = 160$-bits)	$= 1318$ bits (≈ 165 Bytes)
128-bit	($q = 256$-bits)	$= 2086$ bits (≈ 261 Bytes)

Table 5.2: Communication Overhead of A^2-MAKE (*optional)

Architecture	*Communication Overhead*	*Total Size*
PEACE	$2G_1 + 5Z_q$	7q + 2 $= 1192$ bits ($\approx$ 149 Bytes)
A^2-MAKE	$5G_1 + 2Z_q$	7q + 5 $= 1195$ bits ($\approx$ 150 Bytes)

Table 5.3: Comparison of the Communication Overhead (Signature Sizes)

of an elliptic curve point defined over field $\mathbb{F}_q$ as $(q+1)$-bit[21]. In our comparison, we give communication overhead in terms of q (the figures for 170-bit prime q are also provided in the last two columns of Table 5.3). As it is seen from the table, the communication overhead of A^2-MAKE is comparable to the one in PEACE.

A weakness on the underlying group signature scheme of Boneh and Shacham [58] employed by PEACE is mentioned and a corresponding fix demonstrated by Chatterjee et al. [73]. This fix is required to be applied in PEACE due to the fact that, both hashing onto group G_2 together with application of an efficiently computable homomorphism from G_2 to G_1 are required and this necessitates *Type 4* pairing (cf. Section 3.2.1) to be implemented. However, *Type 4* pairing leads to a security weakness in the revocation procedure of the protocol which is shown in [73]. If the proposed solution is accepted as a fix to the base protocol, then computational overhead for the signature generation protocol is no longer $(2P + 8G_1)$ as given in Table 5.1 but $(2P + 6G_1 + 2G_2)$. Furthermore, communication overhead increases to 1533-bits ($\approx$ 192-bytes) since instead of ($5\mathbb{Z}_q$ + $2G_1$) (see Table 5.3), now the corresponding signature consists of ($5\mathbb{Z}_q$ + G_1 + G_2) where a point in G_2 has size equal to 512-bits (cf. Section 3.3.1 of Chatterjee et al. [171]). As a result, our proposed protocol appears to be much more efficient than PEACE and this

[21] Using the point compression methodology explained in [170].

makes our solution more suitable for hybrid wireless mesh networks.

5.6 Implementation and Timing Analysis

In this section, we give the details of our software implementation of the protocols in our framework and provide detailed timings. We utilized the primitives in MIRACL library [172] for the implementation since it is a publicly available library that includes one of the most efficient implementations of both pairing and elliptic curve operations. The protocols are implemented using Visual Studio 2008 and source code is compiled with *Full Optimization (-Ox)* option, which optimizes the code for both speed and size. The platform used to obtain timing results is a PC computer that features a 2.26 GHz Intel(R) Core(TM)2 Duo 32-bit processor with 3GB RAM running Windows operating system. All timings are obtained via Windows-based QueryPerformace functions, and results are given in a resolution of 2,208,066 s^{-1}.

In our implementations, we target two security levels: i) 80-bit as a minimum security level recommended for everyday commercial communications and ii) 128-bit security for sensitive applications. For 80-bit security, we use 160-bit elliptic curves defined over a finite field $\mathbb{F}_q$ where the prime q and the order of the elliptic curve are 160-bit integers. For 128-bit security level, we use 256-bit prime field. In each case, we make use of BN curves [96] where embedding degree is 12. This gives 1920-bit and 3072-bit extension fields for two cases, respectively. The discrete logarithm problem in these extension fields provides sufficient security levels for each case (cf. Table 2 in [71]).

Table 5.4 lists aggregate and individual timing results of protocol steps in our framework for the 80-bit security level. For MAKE protocol, MAKE-UR (User-Router authentication) timings are provided. The same timings for 128-bit security level are given Table 5.5. The timings in Table 5.4 and Table 5.5 represent the average of 10 different simulations. Since aggregate timings include initializations and procedure calls, they are more than the sum of individual timings. On the average, a network user can generate approximately 13 and 5 anonymous signatures per second at 80- and 128-bit security levels, respectively. For 80-bit and 128-bit security levels, it takes approximately 0.49 and

Operation	*SubProtocol Step*	*Party*	*Cost (s)*	*Total Cost (s)*
Setup	1. Complete Protocol Step	STTP		0.078057
Join	2. Complete Protocol Step			0.476978
	2.1. Initialization of Join	NU	0.004708	-
	2.2. Credential Share Generation	NO	0.013305	-
	2.3&4. Credential Generation	STTP	0.029222	-
	2.5. Credential Receive	NU	0.429509	-
MAKE	3. Complete Protocol Step			0.489727
	3.1. Beacon Generation	MR	0.004530	-
	3.2&3. MAKE Part of Network User	NU	-	0.083384
	3.2. Beacon Authentication	NU	0.002987	-
	3.3.a. Key Establishment	NU	0.004491	-
	3.3.b. Anonymous Signature Generation	NU	0.075812	-
	3.4. Verification	MR	-	0.401813
	3.4.c. User RogueList Check	MR	0.000027	-
	3.4.d. Check A' and B'	MR	0.127235	-
	3.4.f. Verify Signature	MR	0.270253	-

Table 5.4: Timing Results of the 160-bit Implementation of A^2-MAKE

1.02 seconds for mutual authentication in MAKE-UR setting, respectively. Table 5.4 and Table 5.5 include also timings for the operations performed by NO and STTP in Join protocol. Join operation is occasionally performed (normally once per user), therefore it is not a bottleneck. We include it just for the record.

Note that individual timings can further be used to approximate timings of MAKE-UU; i.e. user-user authentication protocol whereby a connected user acts as a router; referred as *routing agent* henceforth. As explained earlier, in MAKE-UU, routing agent generates beacon to indicate its availability and verifies the signatures received from other users who want to connect to the network. Here, routing agent acts anonymously since it is also a user who wants to protect his privacy. Therefore, the signature in the beacon must be generated by the same anonymous group signature algorithm used by a regular user for authentication. Since anonymous signature generation is a relatively fast operation, a routing agent can broadcast beacon as frequently as needed. Anonymous signature verification performed by the connecting user to authenticate an anonymous routing agent is slower than generation. It takes the same amount of time as the anonymous signature verification performed by the router in MAKE-UR setting. UserRL check by both sides of

Operation	*SubProtocol Step*	*Party*	*Cost (s)*	*Total Cost (s)*
Setup	1. Complete Protocol Step	STTP		0.143944
Join	2. Complete Protocol Step			0.723288
	2.1. Initialization of Join	NU	0.010379	-
	2.2. Credential Share Generation	NO	0.029816	-
	2.3&4. Credential Generation	STTP	0.062370	-
	2.5. Credential Receive	NU	0.620476	-
MAKE	3. Complete Protocol Step			1.109537
	3.1. Beacon Generation	MR	0.017275	-
	3.2&3. MAKE Part of Network User	NU	-	0.211382
	3.2. Beacon Authentication	NU	0.008859	-
	3.3.a. Key Establishment	NU	0.013373	-
	3.3.b. Anonymous Signature Generation	NU	0.189028	-
	3.4. Verification	MR	-	0.811874
	3.4.c. User RogueList Check	MR	0.000141	-
	3.4.d. Check A' and B'	MR	0.234461	-
	3.4.f. Verify Signature	MR	0.568140	-

Table 5.5: Timing Results of the 256-bit Implementation of A^2-MAKE

MAKE-UU can be delegated to a more powerful user after the connection is established, in case UserRL is not available to the parties or it is too big to perform this check efficiently. If either side of the connection turns out to be a revoked user, then the connection can be terminated immediately.

In Tables 5.4 and 5.5, timing results for the *UserRL check* are given for the initial case, i.e. when there are no revoked users in the network. However, as network starts serving, the number of users in UserRL is likely to increase. In Table 5.6, timings for the UserRL checking (see Step 4c of MAKE-UR) algorithm are listed for the cases where there are 1, 10, 50, 100 and 200 users in UserRL for both 160-bit and 256-bit key sizes.

Since network users in our framework are likely to possess resource-constrained devices in terms of computation, battery power, etc., the most time-critical steps of our protocol are the ones that are performed by the network user. In order to assess the computational burden on a network user, detailed timings for the steps taken by NU are given in Table 5.7, where results for 256-bit key size are put in parenthesis.

As can be observed from Table 5.7, a network user needs only moderate time to connect to the network anonymously. Therefore, A^2-MAKE protocol is feasible even for the

Operation	*SubProtocol Step*	*Number Of Rogue Users*	*Cost (s)*
MAKE	3.4. Verification	-	-
	3.4.c. User RogueList Check	-	-
	Key Size: 160 bit	1	0.0021323
		10	0.0202548
		50	0.1049326
		100	0.2057378
		200	0.3865578
	Key Size: 256 bit	1	0.0034626
		10	0.0490993
		50	0.2590574
		100	0.4295544
		200	0.7920474

Table 5.6: Time Costs of UserRL Checking for 1, 10, 50, 100 and 200 Rogue Users

resource-constrained devices to authenticate themselves anonymously in wireless mesh networks.

5.6.1 Timing Results for a Resource Constrained User

In order to see the performance of the protocol on a relatively low-end computing device, we scale down our timing results for Intel® Atom™ Processor Z500. It is an embedded processor targeted for Netbooks, nettops, and Mobile Internet Devices (MIDs) with a modest 800 MHz clock frequency and 512 KB cache memory. Here, we provide the cost related to the Network User itself in Table 5.8. For even more constrained devices, a network user can securely delegate some part of pairing computation to a more powerful entity in the network as suggested in [173].

5.7 Simulation Results

We conducted some experiments on ns-3 (version 13) [174], on Ubuntu 10.04 platform to show the efficiency of the proposed protocol on a real life like scenario. Since ns-3 is a discrete event simulator, system properties of the computer on which the simulations are made do not have any effect on the results.

Operation	*SubProtocol Step*	*Cost (s)*	*Total Cost (s)*
Join	(2.) Total Time Spent in Join		0.434217 (0.630855)
	2.1. Initialization of Join	0.004708 (0.010379)	- -
	2.5. Credential Receive	0.429509 (0.620476)	- -
MAKE	(3.) Total Time Spent in MAKE	- -	0.083290 (0.211260)
	3.2. Beacon Authentication	0.002987 (0.008859)	- -
	3.3. AKE Part of Network User	- -	0.080303 (0.202401)
	3.3.a. Key Establishment	0.004491 (0.013373)	- -
	3.3.b. Anonymous Signature Generation	0.075812 (0.189028)	- -

Table 5.7: Detailed Timings for the Protocol Steps taken by the Network User

Operation	*SubProtocol Step*	*Cost (s)*	*Total Cost (s)*
Join	(2.) Total Time Spent in Join		1.198439 (1.741160)
	2.1. Initialization of Join	0.012994 (0.028646)	- -
	2.5. Credential Receive	1.185445 (1.712514)	- -
MAKE	(3.) Total Time Spent in MAKE	- -	0.229880 (0.583078)
	3.2. Beacon Authentication	0.008244 (0.024451)	- -
	3.3. AKE Part of Network User	- -	0.221636 (0.558627)
	3.3.a. Key Establishment	0.012395 (0.036909)	- -
	3.3.b. Anonymous Signature Generation	0.209241 (0.521717)	- -

Table 5.8: Detailed Timings for the Protocol Steps executed by the Network User on an Embedded Processor

In all our simulations, the simulated nodes are placed in a 4000m $\times$ 4000m square shape area. The number of mesh clients in simulations varies between 50 to 300 by 50 increments. Furthermore, the number of routers is taken as 121. The routers are placed at fixed positions on a grid in the network simulation area, and thus the distance between routers is 400 meters. The mesh clients start their movements at random points within the area and do random movements within it. The randomness for the users' movements is obtained by the random path generation algorithm provided in ns-3.13. Packet queue size of mesh routers and relaying mesh clients is assumed to be constant, which is set to 10 packets in our simulations, meaning that some of the packets will be dropped if the queue is full. Therefore, increased number of packets causes an increase in the rate of dropped packets.

In our simulations, 30% of the users are assumed to act as routers, i.e. relaying network users (or agents), and used by normal users as a relaying agent to authenticate themselves and gain access to the network and related services. Relaying users in this network are not assumed to be a part of the network backbone. Unlike the network operator and mesh routers, they have to authenticate with a router first in order to connect to the network and then perform the relaying activity.

All routers are assumed to be informed instantly by the network administrator of the updated revocation list (UserRL) using the established network. On the other hand, mesh clients that are acting as relaying agents obtain this updated list from a router only if they are connected to the network. This creates a traffic on the wireless network. These updates are assumed to be broadcast to corresponding receivers at three different time intervals; 60, 180, and 300 seconds. Furthermore, in every 30 seconds, routers broadcast their public parameters together with a signature, the beacon, to all users in vicinity. In addition, if there are any relaying users connected to the routers, they also broadcast their public parameters along with an anonymous group signature in every 30 seconds. All of the simulations were performed for one-day of simulated time.

In these simulations, it is assumed that mesh clients, either relaying agent or a normal user, are running the protocol steps on a processor with 800 MHz clock frequency (i.e. timings are taken for the platform with Atom$^{\text{TM}}$ Processor Z500). On the other hand,

mesh routers are assumed to be running on a processor similar to the one used in protocol implementations, a dual core 2.26 GHz processor. As a result, at 80-bit security level, anonymous signatures generated by the mesh clients are verified by the corresponding mesh routers in 0.4018 s, whereas the verification is completed in 1.109 s by a relaying agent. On the other hand, it is assumed that the verification of mesh router's conventional signature by the corresponding client together with the generation of an anonymous signature is accomplished in 0.2299 s. On the other hand, the verification is completed in 1.319 s if a mesh client tries to connect to the network through a relaying agent and verifies the anonymous signature received from her and generates its own anonymous signature required for authentication.

Similarly, for the simulations performed at 128-bit security level, corresponding timings used are, 0.8119 s for the verification of anonymous signatures by a mesh router and 2.241 s by relaying mesh client. Verification and anonymous signature generation by the mesh client take 0.5831 s if a mesh router is the authenticator, whereas it takes 2.774 s when the authenticator is a relaying mesh client.

We perform our simulations on two different scenarios based on where the UserRL is held. In the first scenario, it is assumed that UserRL is held by mesh clients in addition to the mesh routers. On the other hand, in the second scenario, UserRL is only held by the mesh routers. A relaying mesh client asks the router it is connected, to perform UserRL checking for another client which she assists to connect to the network. In both scenarios, we examine the authentication times and the number of successful connections made. In the first scenario, differing from the second one, we analyze the number of true positive authentications made by the relaying mesh clients. True positive authentication is the ratio of the number of authentications accomplished by the relaying mesh clients with the updated UserRL to the total successful authentications made by her throughout the lifetime of the network including the authentications made with obsolete UserRL.

5.7.1 Scenario 1: UserRL is held both at mesh routers and mesh clients

In this section, results of the simulations performed considering the three different UserRL broadcast time intervals are analyzed. In this current scenario, where mesh clients hold UserRL locally, time intervals are assumed to be 60, 180, and 300 seconds between each UserRL broadcast.

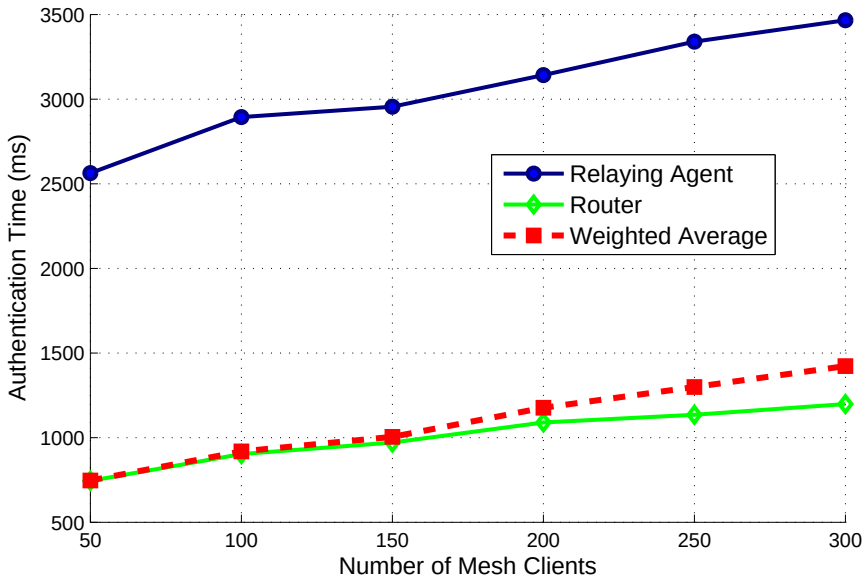

Figure 5.2: Authentication Times at 80-bit Security Level

Figure 5.2 shows the average authentication time of the mesh clients with respect to the number of the mesh clients within the network at 80-bit security level. Figure 5.3 similarly shows the average authentication time of the mesh clients at 128-bit security level. Average time of the authentications made by mesh routers and relaying mesh clients are shown separately together with a weighted average of them. The average of all timings obtained from three different simulations corresponding to the three different UserRL broadcast time intervals are given as the authentication time. Weighted average is calculated by dividing the total time spent on all successful authentications performed by both

parties by the total number of successful authentications.

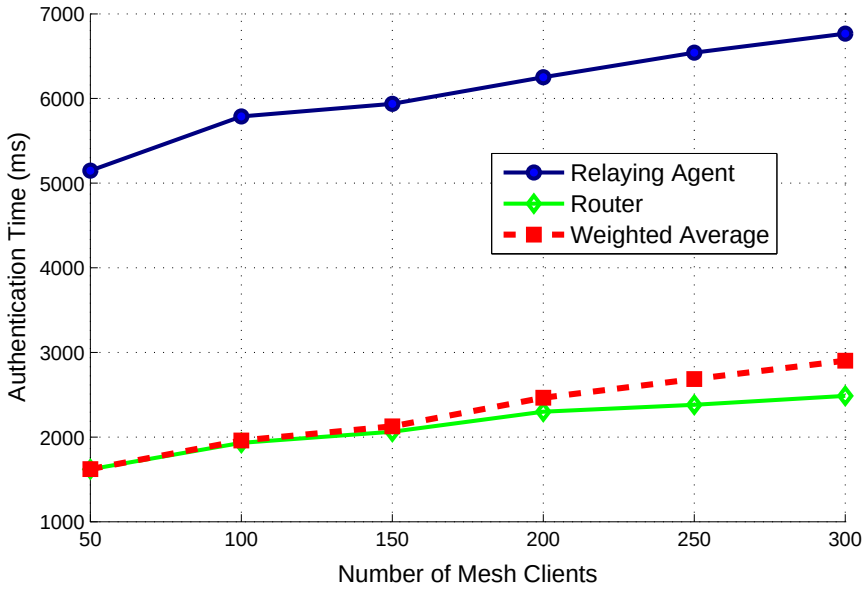

Figure 5.3: Authentication Times at 128-bit Security Level

As it is seen from the Figure 5.2 and Figure 5.3, ceteris paribus, average authentication time increases linearly with the increasing number of mesh clients. However, average authentication time increases very slowly as the number of mesh clients increases. Weighted average authentication time increases approximately 85%, and 75% at most at 80-bit and 128-bit security levels, respectively, with respect to a six fold increase in the number of mesh clients.

Number of successful authentications made by relaying mesh clients and routers at 80-bit security level is given in Figure 5.4. The results are similar for 128-bit security level. These numbers are used in the calculation of the weighted authentication time and explain why the weighted authentication time in Figures 5.2 and 5.3 is nearly the same as the average authentication time resulting from the operation performed by the mesh routers. The latter is due to the fact that, on the average, approximately the 95% of all the authentications are accomplished by the mesh routers. Furthermore, the total number of

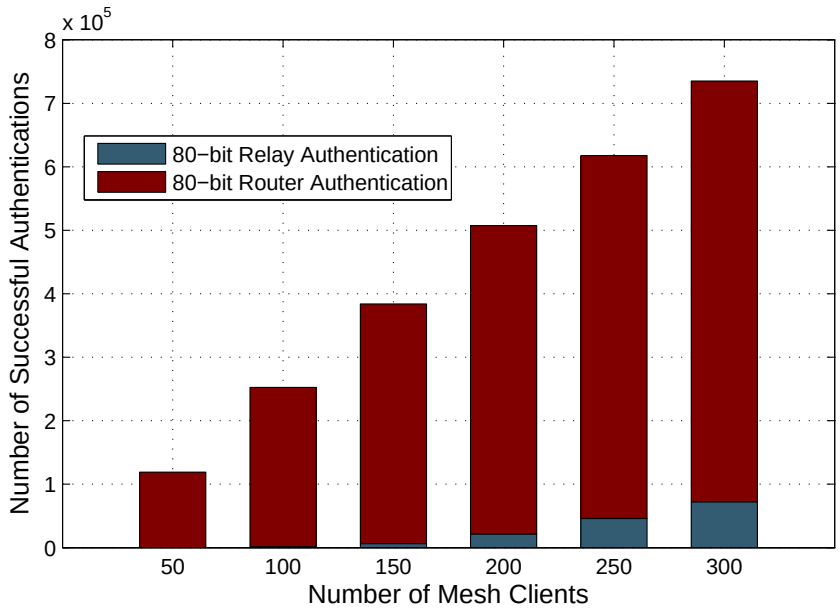

Figure 5.4: Number of Successful Authentications by Routers and Relaying Agents

successful authentications made increases linearly with respect to increasing number of mesh clients as expected.

Another important metric is the ratio of successful authentication attempts. This metric is calculated as ratio of the number of successful authentications to the number of authentication requests made. Figure 5.5 demonstrates the ratio of weighted average of the successful authentication attempts at 80-bit and 128-bit security levels. This ratio decreases with the increasing number of mesh clients. This is expected, since the number of packets throughout the network increases with the increasing number of mesh clients, whereas the number of mesh routers stays constant. Furthermore, each mesh router and relaying mesh client can handle only limited number of packets.

Moreover, Figure 5.6 gives these ratios for the successful authentications made by mesh routers and relaying mesh clients separately. As it is seen from Figure 5.6, ratio drops from nearly 0.92 to 0.70 for the authentication attempts made to the relaying agents as number of mesh clients increases from 50 to 300. On the other hand, a decrease in the ratio is also observed for the authentication attempts made to the mesh routers while it is

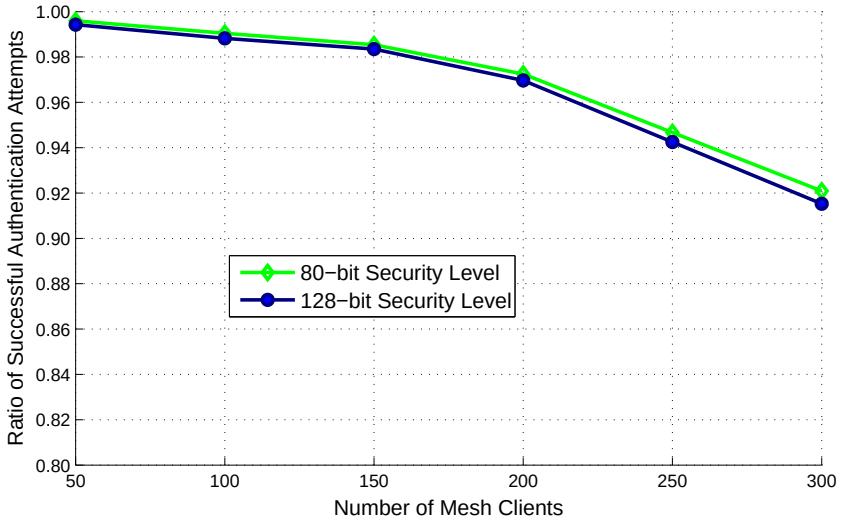

Figure 5.5: Ratio of Successful Authentication Attempts (Weighted average of Relaying agent and Router Authentications)

not as steep.

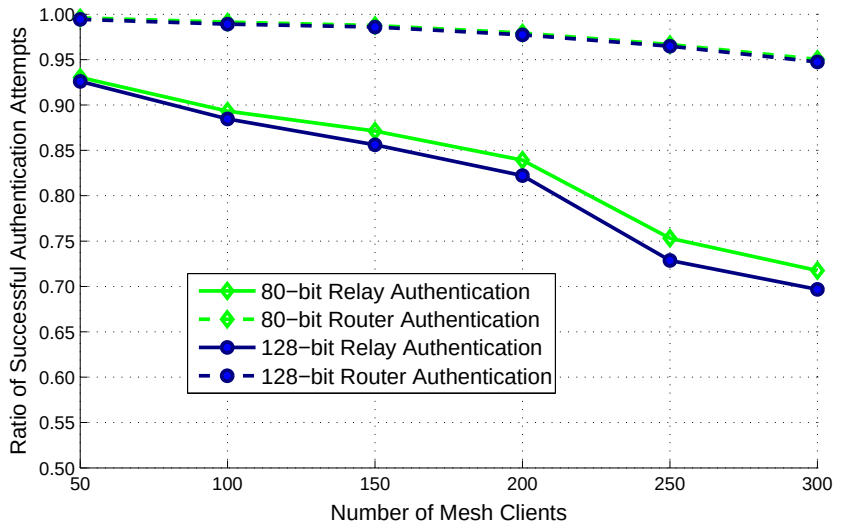

Figure 5.6: Ratio of Successful Authentication Attempts (Relaying agent and Router Authentications are shown separately)

Authentication of mesh clients are performed by the mesh routers and relaying agents where all these authenticators perform UserRL checking locally. Although the mesh routers are informed instantly by the network administrator for the updated UserRL, relaying agents are not able to obtain the updated list if they are not connected to the network during UserRL broadcast. As a result, it is possible for a relaying mesh client to perform authentication with an obsolete UserRL. We call the authentications made by relaying mesh clients with the updated UserRL as true positive authentications. In Figure 5.7, ratio of the true positive authentications made by the relaying agents to the total number of authentications is given. As it is seen from the Figure 5.7, generally true positive ratio

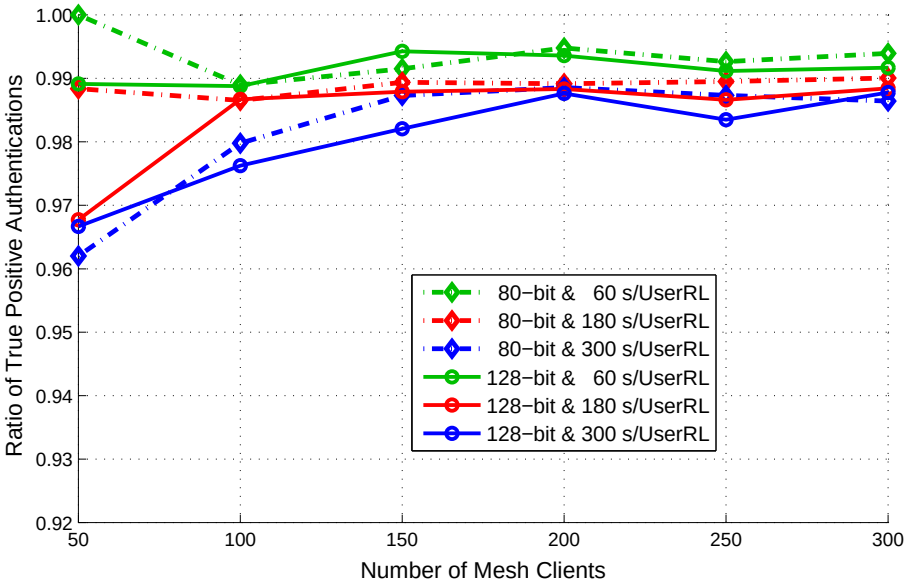

Figure 5.7: True Positive Authentications made by Relaying Mesh Clients

decreases with the increasing UserRL broadcast time interval. However, this behaviour loosens with the increasing number of mesh clients within the network. Furthermore, security level does not seem to have a meaningful impact on this ratio.

5.7.2 Scenario 2: UserRL is held only at mesh routers

In this scenario, it is assumed that UserRL is held only at mesh routers and relaying mesh clients do not have access to them. As a result, in order to authenticate another mesh client, relaying agent sends data values used in UserRL checking to the mesh router it is already connected to, and asks this router to perform UserRL checking. In simulations, it is assumed that there are 10 clients in the list throughout the simulated time. Therefore, it is assumed that the mesh routers perform UserRL checking in 0.02026 s, and 0.04909 s for 80-bit and 128-bit security levels, respectively.

Figure 5.8 and Figure 5.9 show the authentication time of the mesh clients at 80-bit and 128-bit security levels, respectively.

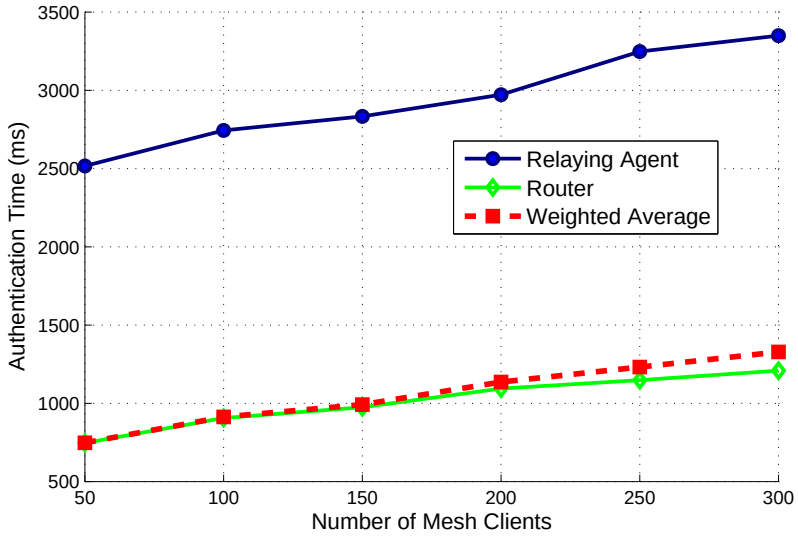

Figure 5.8: Authentication Times at 80-bit Security Level

Similar to the results obtained from the simulations performed for the first scenario, average authentication time increases linearly with the increasing number of mesh clients. It increases very slowly as the number of mesh clients increases. Weighted average authentication time increases approximately 75%, and 65% at most at 80-bit and 128-bit

security levels, respectively, with respect to a six fold increase in the number of mesh clients. Related figure is the number of successful authentications made by relaying mesh

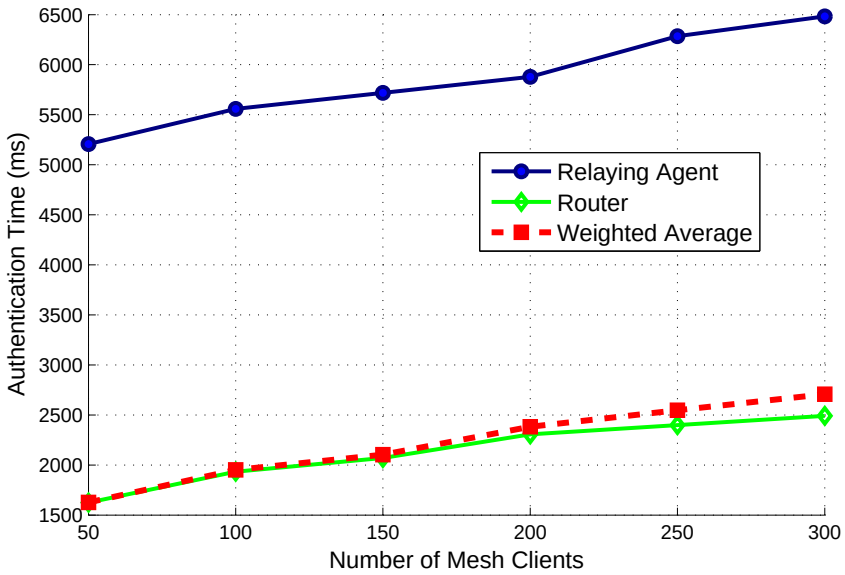

Figure 5.9: Authentication Times at 128-bit Security Level

clients and router. Figure 5.10 shows the corresponding results both at 80-bit security level. The results are similar for 128-bit security level.

The ratio of number of successful authentications to the number connection attempts made for the second scenario is given in Figure 5.11. In addition, Figure 5.12 demonstrates the corresponding ratio for the authentications made by the relaying mesh clients and mesh routers.

Comparing Figure 5.12 with corresponding Figure 5.6, it is seen that the ratio of the successful authentications is lower for the second scenario where the UserRL checking is performed only by the mesh routers. This difference is notable in authentications made by the relaying mesh clients. This may be due to the increased packet drops throughout the network and increased response time of the mesh routers to the UserRL checking requests.

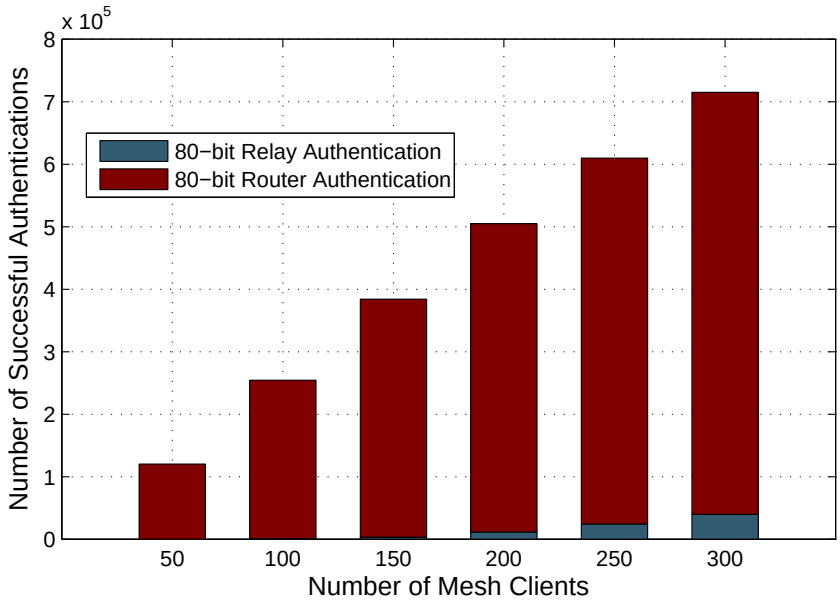

Figure 5.10: Number of Successful Authentications by Routers and Relaying Agents

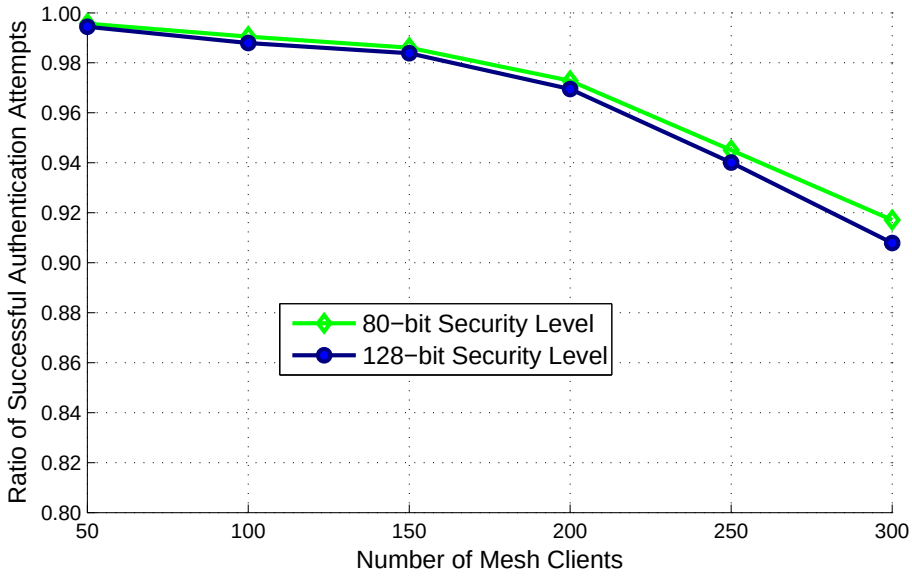

Figure 5.11: Ratio of Successful Authentication Attempts (Weighted average of Relaying agent and Router Authentications)

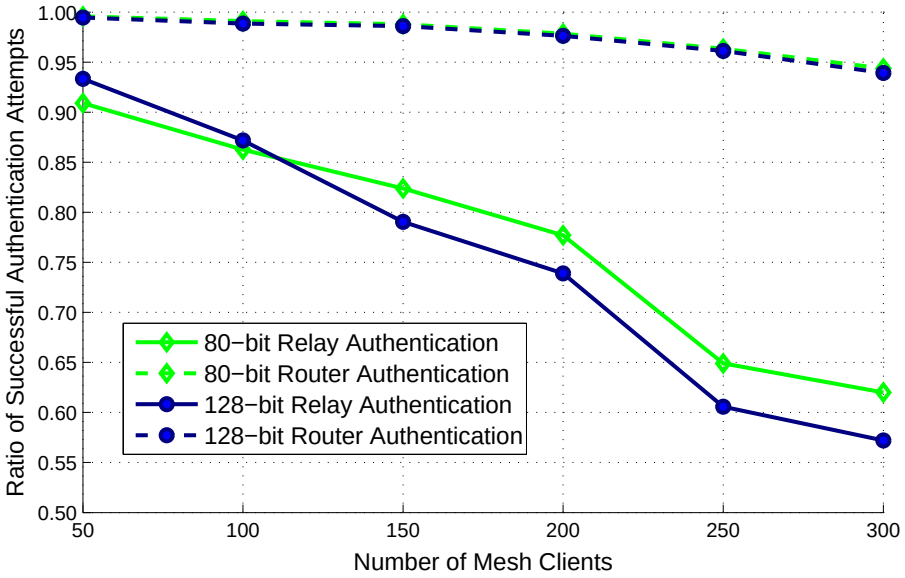

Figure 5.12: Ratio of Successful Authentication Attempts (Relaying agent and Router Authentications are shown separately)

As a result, authentication times obtained from the simulations performed for this scenario are mostly lower than the ones obtained in the first scenario. This may occur since the authentications that require more time are possibly dropped, either at the router due to the packet queue being full or within the network, leaving successful attempts having comparatively lower authentication times. This possibly compensates the expected increase in authentication times due to relaying agents waiting acknowledgements for the UserRL checking requests.

Lastly, ratio of true positive authentications is 1.0 in this scenario. This is due to the fact that relaying mesh clients always delegate UserRL checking to mesh routers that possess the updated UserRL.

Chapter 6

Concluding Remarks

The proposed framework A^2-MAKE herein empowers wireless mesh network users with a secure, privacy-preserving authentication and related protocols while allowing the network owner to implement user accountability. Due to sophisticated yet efficient signature generation and verification algorithms, the proposed protocols are well suited to WMNs, where resource constrained devices perform routing and configuration activities. These algorithms are derived from a proposal made for an advanced application of group signatures known as direct anonymous attestation, which is the pillar of our framework. The framework allows registered users to connect to the network anonymously when a network router or a relaying agent is available within the communication range. The proposed framework provides strong user privacy (both anonymity and unlinkability) and user accountability, both of which have not been provided together by the proposals in the current literature.

The primary contribution of this thesis is is a framework for wireless mesh (or similar) networks that provide efficient and applicable solutions to the security, privacy and trust requirements. The proposed solutions protect the privacy and security of the users within such networks, not only against the adversaries or other users but also against powerful entities such as network operators. While protecting privacy of network users is of utmost importance, accountability for irresponsible and malicious user behavior can also be efficiently implemented in the proposed framework. Eliminating a single, powerful entity that has the power of violating users' privacy by using a sharing mechanism distinguishes this work from the previous solutions.

At the technical level, the main contribution of this thesis is the three-party Join protocol together with two-party user identification and revocation protocols that reconcile the user privacy and user accountability in an efficient and scalable manner. Our framework offers two efficient and scalable algorithms, user identification and revocation, whereby user identities and private keys can be recovered in a controlled manner. User identification procedure is separated from the revocation procedure in order to allow for proper investigation of a suspected but innocent user without revoking her key. This is made possible only through the collaboration of two semi-trusted parties, namely the STTP and the NO; therefore nobody can violate the privacy of users alone. Revocation procedure is under the control of the STTP, which is assumed to behave as described in the protocol steps. User revocation is obtained by adding the private key of a revoked user to the user revocation list.

Security analysis for the proposed framework is also provided, in which the security and privacy of the protocols are reduced to well-known computationally hard problems. The assumptions on powerful entities, such as network operator and trusted third party, are relaxed since they do not have to be fully trusted as required in previous works. In our framework, they are semi-trusted and non-colluding; two properties which are common in cryptographic settings and easier to implement in practice. In addition, backward security and privacy are provided for revoked users.

Computational and communication performances of signature generation and verification protocols in comparison with a similar protocol in literature are analyzed. As a result, it is shown that our protocol outperforms a similar protocol from the efficiency point of view. Furthermore, user identification and revocation can be performed efficiently and the algorithms used in these procedures scale well with the number of users.

Protocols in the proposed framework are implemented at different levels of security and resulting timings are given for a typical desktop computer. Approximate timings for constrained devices are provided as well. In addition, mesh network simulations are performed in order to evaluate the actual costs pertaining to the proposed procedures including network related losses. Implementation and simulation results demonstrate that the framework can be practically deployed in hybrid wireless mesh networks to address

security, privacy and accountability concerns effectively. Since the protocols are generic, applications that require anonymous authentication within other types of networks can also benefit from the proposed framework.

The framework can easily be extended to accommodate advanced features such as user groups and role-based access. Additionally, incorporating the advanced features developed in this thesis into Cloud Computing and other network types such as Vehicular Networks is left as a future research. Also, distribution of user identification to prevent certain attack types (e.g. sybil attacks) to designated entities in the network is another research avenue to facilitate faster user identification and revocation processes.

Bibliography

[1] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups (extended abstract). In *Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*, pages 410–424, London, UK, 1997. Springer-Verlag. ISBN 3-540-63384-7.

[2] Atsuko Miyaji, Masaki Nakabayashi, and Shunzou Takano. New explicit conditions of elliptic curve traces for fr-reduction. In *IEICE Transactions on Fundamentals, E84-A(5)*, pages 1234–1243, 2001.

[3] Cécile Delerablée and David Pointcheval. Dynamic fully anonymous short group signatures. In Phong Q. Nguyen, editor, *VIETCRYPT*, volume 4341 of *Lecture Notes in Computer Science*, pages 193–210. Springer, 2006. ISBN 3-540-68799-8.

[4] Kui Ren and Wenjing Lou. A sophisticated privacy-enhanced yet accountable security framework for metropolitan wireless mesh networks. In *ICDCS*, pages 286–294. IEEE Computer Society, 2008. ISBN 978-0-7695-3172-4.

[5] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[6] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21:120–126, February 1978. ISSN 0001-0782.

[7] Taher El Gamal. A public key cryptosystem and a signature scheme based on

discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 10–18, New York, NY, USA, 1985. Springer-Verlag New York, Inc. ISBN 0-387-15658-5.

[8] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17:281–308, April 1988. ISSN 0097-5397.

[9] David Chaum and Eugène van Heyst. Group signatures. In *EUROCRYPT*, pages 257–265, 1991.

[10] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions. In *Proceedings of the 22nd international conference on Theory and applications of cryptographic techniques*, EUROCRYPT'03, pages 614–629, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-14039-5.

[11] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In Alfred Menezes, editor, *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 136–153. Springer, 2005. ISBN 3-540-24399-2.

[12] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in cryptology*, pages 554–567. Springer-Verlag, 2001.

[13] Lingling Wang, Guoyin Zhang, and Chunguang Ma. A survey of ring signature. *Frontiers of Electrical and Electronic Engineering in China*, 3:10–19, 2008. ISSN 1673-3460.

[14] Ian F. Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, 2005.

[15] John R. Douceur. The sybil attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 251–260, London, UK, 2002. Springer-Verlag. ISBN 3-540-44179-4.

[16] L Chen, P. Morrissey, and N.P. Smart. DAA: Fixing the pairing based protocols. Cryptology ePrint Archive, Report 2009/198, 2009. `http://eprint.iacr.org/`.

[17] Stefan A. Brands. An efficient off-line electronic cash system based on the representation problem. Technical report, CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands, The Netherlands, 1993.

[18] Dan Boneh. The decision diffie-hellman problem. In *Proceedings of the Third International Symposium on Algorithmic Number Theory*, pages 48–63, London, UK, 1998. Springer-Verlag. ISBN 3-540-64657-4.

[19] Markus Stadler. Publicly verifiable secret sharing. In *Proceedings of the 15th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'96, pages 190–199, Berlin, Heidelberg, 1996. Springer-Verlag. ISBN 3-540-61186-X.

[20] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In Howard M. Heys and Carlisle M. Adams, editors, *Selected Areas in Cryptography*, volume 1758 of *Lecture Notes in Computer Science*, pages 184–199. Springer, 1999. ISBN 3-540-67185-4.

[21] Ronald L. Rivest and Burt Kaliski. Rsa problem. In Henk C. A. van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security (2nd Ed.)*, pages 1065–1069. Springer, 2011. ISBN 978-1-4419-5905-8.

[22] Niko Baric and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Proceedings of the 16th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'97,

pages 480–494, Berlin, Heidelberg, 1997. Springer-Verlag. ISBN 3-540-62975-0.

[23] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*, pages 16–30, London, UK, 1997. Springer-Verlag. ISBN 3-540-63384-7.

[24] Ronald Cramer and Victor Shoup. Signature schemes based on the strong rsa assumption. In *Proceedings of the 6th ACM conference on Computer and communications security*, CCS '99, pages 46–51, New York, NY, USA, 1999. ACM. ISBN 1-58113-148-8.

[25] Jan Camenisch and Markus Michels. A group signature scheme with improved efficiency. In *Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '98, pages 160–174, London, UK, 1998. Springer-Verlag. ISBN 3-540-65109-8.

[26] Jan Camenisch and Markus Michels. A group signature scheme based on an rsa-variant. Technical report, BRICS, 1998.

[27] Amos Fiat and Adi Shamir. How to prove yourself: practical solutions to identification and signature problems. In *Proceedings on Advances in cryptology—CRYPTO '86*, pages 186–194, London, UK, 1987. Springer-Verlag. ISBN 0-387-18047-8.

[28] U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *J. Cryptol.*, 1:77–94, August 1988. ISSN 0933-2790.

[29] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, CCS '93, pages 62–73, New York, NY, USA, 1993. ACM. ISBN 0-89791-629-8.

[30] David Pointcheval and Jacques Stern. Security proofs for signature schemes. In

Proceedings of the 15th annual international conference on Theory and application of cryptographic techniques, EUROCRYPT'96, pages 387–398, Berlin, Heidelberg, 1996. Springer-Verlag. ISBN 3-540-61186-X.

[31] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51:557–594, July 2004. ISSN 0004-5411.

[32] Claus P. Schnorr. Efficient identification and signatures for smart cards. In *Proceedings on Advances in cryptology*, CRYPTO '89, pages 239–252, New York, NY, USA, 1989. Springer-Verlag New York, Inc. ISBN 0-387-97317-6.

[33] David Chaum, Jan-Hendrik Evertse, and Jeroen Van De Graaf. An improved protocol for demonstrating possession of discrete logarithms and some generalizations. In *Proceedings of the 6th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'87, pages 127–141, Berlin, Heidelberg, 1988. Springer-Verlag. ISBN 3-540-19102-X.

[34] Ivan Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *Proceedings of the 19th international conference on Theory and application of cryptographic techniques*, EUROCRYPT'00, pages 418–430, Berlin, Heidelberg, 2000. Springer-Verlag. ISBN 3-540-67517-5.

[35] David Chaum. Zero-knowledge undeniable signatures (extended abstract). In *Proceedings of the workshop on the theory and application of cryptographic techniques on Advances in cryptology*, EUROCRYPT '90, pages 458–464, New York, NY, USA, 1991. Springer-Verlag New York, Inc. ISBN 0-387-53587-X.

[36] David Chaum and Torben P. Pedersen. Wallet databases with observers. In *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '92, pages 89–105, London, UK, 1993. Springer-Verlag. ISBN 3-540-57340-2.

[37] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowl-

edge and simplified design of witness hiding protocols. In *Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '94, pages 174–187, London, UK, 1994. Springer-Verlag. ISBN 3-540-58333-5.

[38] Jan Camenisch. Efficient and generalized group signatures. In *Proceedings of the 16th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'97, pages 465–479, Berlin, Heidelberg, 1997. Springer-Verlag. ISBN 3-540-62975-0.

[39] Anna Lysyanskaya and Zulfikar Ramzan. Group blind digital signatures: A scalable solution to electronic cash. In *Proceedings of the Second International Conference on Financial Cryptography*, pages 184–197, London, UK, 1998. Springer-Verlag. ISBN 3-540-64951-4.

[40] Joe Kilian and Erez Petrank. Identity escrow. In *Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology*, pages 169–185, London, UK, 1998. Springer-Verlag. ISBN 3-540-64892-5.

[41] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177): 203–209, January 1987. ISSN 0025-5718.

[42] Victor S. Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology*, CRYPTO '85, pages 417–426, London, UK, UK, 1986. Springer-Verlag. ISBN 3-540-16463-4.

[43] J. H. Silverman. *The Arithmetic of Elliptic Curves*. Springer-Verlag, Berlin, 1995.

[44] Ian F. Blake, G. Seroussi, and N. P. Smart. *Elliptic curves in cryptography*. Cambridge University Press, New York, NY, USA, 1999. ISBN 0-521-65374-6.

[45] Lawrence C. Washington. *Elliptic Curves: Number Theory and Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 2003. ISBN 1584883650.

[46] William C. Waterhouse. Abelian varieties over finite fields. *Ann. Scient. Ec. Norm. Sup.*, 4(2):521–560, 1969.

[47] Hans-Georg Ruck. A note on elliptic curves over finite fields. *Mathematics of Computation*, 49(179):301–304, 1987.

[48] S. Pohlig and M. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (Corresp.). *Information Theory, IEEE Transactions on*, 24(1):106–110, 1978.

[49] Darrel Hankerson, Alfred J. Menezes, and Scott Vanstone. *Guide to Elliptic Curve Cryptography*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003. ISBN 038795273X.

[50] Nigel P. Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of Cryptology*, 12:193–196, 1999.

[51] Alfred Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.

[52] Gerhard Frey and Hans-Georg Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Math. Comput.*, 62:865–874, April 1994. ISSN 0025-5718.

[53] Arjen K. Lenstra and Eric R. Verheul. Selecting cryptographic key sizes. *Journal of Cryptology*, 14:255–293, 1999.

[54] Antoine Joux. A one round protocol for tripartite diffie-hellman. In Wieb Bosma, editor, *ANTS*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394. Springer, 2000. ISBN 3-540-67695-3.

[55] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In Kilian [175], pages 213–229. ISBN 3-540-42456-3.

[56] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53, New York, NY, USA, 1985. Springer-Verlag New York, Inc. ISBN 0-387-15658-5.

[57] Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems based on pairing. In *Proceedings of the Symposium on Cryptography and Information Security, Okinawa, Japan*, pages 26–28, January, 2000.

[58] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In Atluri et al. [176], pages 168–177. ISBN 1-58113-961-6.

[59] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Franklin [177], pages 41–55. ISBN 3-540-22668-0.

[60] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Franklin [177], pages 56–72. ISBN 3-540-22668-0.

[61] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *EUROCRYPT*, pages 506–522, 2004.

[62] S. Mitsunari, R. Sakai, and M. Kasahara. A new traitor tracing. *EICE Trans Fundam Electron Commun Comput Sci (Inst Electron Inf Commun Eng)*, E85-A: 481–484, 2002. ISSN 0916-8508.

[63] Sherman S. M. Chow, Lucas Chi Kwong Hui, and Siu-Ming Yiu. Identity based threshold ring signature. In Choonsik Park and Seongtaek Chee, editors, *ICISC*, volume 3506 of *Lecture Notes in Computer Science*, pages 218–232. Springer, 2004. ISBN 3-540-26226-1.

[64] Ernie Brickell, Liqun Chen, and Jiangtao Li. A new direct anonymous attestation scheme from bilinear maps. In Peter Lipp, Ahmad-Reza Sadeghi, and Klaus-Michael Koch, editors, *TRUST*, volume 4968 of *Lecture Notes in Computer Science*, pages 166–178. Springer, 2008. ISBN 978-3-540-68978-2.

[65] Xiaofeng Chen and Dengguo Feng. Direct anonymous attestation for next generation tpm. *JCP*, 3(12):43–50, 2008.

[66] Liqun Chen, Paul Morrissey, and Nigel P. Smart. Pairings in trusted computing. In Galbraith and Paterson [178], pages 1–17. ISBN 978-3-540-85503-3.

[67] Liqun Chen. A daa scheme requiring less tpm resources. *IACR Cryptology ePrint Archive*, 2010:8, 2010.

[68] Ernie Brickell and Jiangtao Li. A pairing-based daa scheme further reducing tpm resources. In Acquisti et al. [179], pages 181–195. ISBN 978-3-642-13868-3.

[69] Liqun Chen, Dan Page, and Nigel P. Smart. On the design and implementation of an efficient daa scheme. In Gollmann et al. [180], pages 223–237. ISBN 978-3-642-12509-6.

[70] Hovav Shacham. *New Paradigms in Signature Schemes*. PhD thesis, Stanford University, California, December 2005.

[71] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[72] L. Chen, Z. Cheng, and N. P. Smart. Identity-based key agreement protocols from pairings. *Int. J. Inf. Secur.*, 6:213–241, June 2007. ISSN 1615-5262.

[73] Sanjit Chatterjee, Darrel Hankerson, and Alfred Menezes. On the efficiency and security of pairing-based protocols in the type 1 and type 4 settings. In M. Anwar Hasan and Tor Helleseth, editors, *WAIFI*, volume 6087 of *Lecture Notes in Computer Science*, pages 114–134. Springer, 2010. ISBN 978-3-642-13796-9.

[74] Ben Lynn. *On the implementation of pairing-based cryptosystems*. PhD thesis, Stanford University, California, 2007.

[75] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in*

Computer Science, pages 56–73. Berlin: Springer-Verlag, 2004. Available at `http://www.cs.stanford.edu/~xb/eurocrypt04a/`.

[76] Dan Boneh and Xavier Boyen. Short signatures without random oracles and the sdh assumption in bilinear groups. *J. Cryptol.*, 21:149–177, February 2008. ISSN 0933-2790.

[77] Paulo S. L. M. Barreto, Steven Galbraith, Colm O Heigeartaigh, and Michael Scott. Efficient pairing computation on supersingular abelian varieties. In *Designs, Codes and Cryptography*, pages 239–271, 2004.

[78] Florian Hess, Nigel P. Smart, and Frederik Vercauteren. The eta pairing revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006.

[79] Florian Hess. Pairing lattices. In Galbraith and Paterson [178], pages 18–38. ISBN 978-3-540-85503-3.

[80] Victor S. Miller. The weil pairing, and its efficient calculation. *J. Cryptol.*, 17: 235–261, September 2004. ISSN 0933-2790.

[81] André Weil. Sur les fonctions algebriques à corps de constantes finis. *C.R.Acad.Sci.Paris*, 210:592–594, 1940.

[82] Victor S. Miller. Short programs for functions on curves. In *IBM Thomas J. Watson Research Center*, 1986.

[83] John Tate. Wc-groups over p-adic fields. *Seminaire Bourbaki, expose 156, Secretariat mathematique, Paris*, decembre 1957.

[84] Stephen Lichtenbaum. Duality theorems for curves over p-adic fields. *Inventiones math.*, 7:120–136, 1969.

[85] Gerhard Frey, Michael Müller, and Hans-Georg Rück. The tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. *IEEE Transactions on Information Theory*, 45(5):1717–1719, 1999.

[86] Eunjeong Lee, Hyang-Sook Lee, and Cheol-Min Park. Efficient and generalized pairing computation on abelian varieties. *IEEE Trans. Inf. Theor.*, 55:1793–1803, April 2009. ISSN 0018-9448.

[87] Seiichi Matsuda, Naoki Kanayama, Florian Hess, and Eiji Okamoto. Optimised versions of the ate and twisted ate pairings. In *the Eleventh IMA International Conference on Cryptography and Coding*, pages 302–312. Springer-Verlag, 2007.

[88] Chang-An Zhao, Fangguo Zhang, and Jiwu Huang. A note on the ate pairing. *International Journal of Information Security*, 7:379–382, 2008. ISSN 1615-5262. 10.1007/s10207-008-0054-1.

[89] I. Blake, G. Seroussi, N. Smart, and J. W. S. Cassels. *Advances in Elliptic Curve Cryptography (London Mathematical Society Lecture Note Series)*. Cambridge University Press, New York, NY, USA, 2005. ISBN 052160415X.

[90] Don Coppersmith. Fast evaluation of logarithms in fields of characteristic two. *IEEE Transactions on Information Theory*, 30(4):587–593, 1984.

[91] Leonard M. Adleman. The function field sieve. In *Proceedings of the First International Symposium on Algorithmic Number Theory*, pages 108–121, London, UK, 1994. Springer-Verlag. ISBN 3-540-58691-1.

[92] Leonard M. Adleman and Ming-Deh A. Huang. Function field sieve method for discrete logarithms over finite fields. *Information and Computation*, 151(1-2):5 – 16, 1999. ISSN 0890-5401.

[93] Antoine Joux and Reynald Lercier. The function field sieve is quite special. In *Proceedings of the 5th International Symposium on Algorithmic Number Theory*, ANTS-V, pages 431–445, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-43863-7.

[94] Eric R. Verheul. Evidence that xtr is more secure than supersingular elliptic curve cryptosystems. *J. Cryptology*, 17(4):277–296, 2004.

[95] Friederike Brezing and Annegret Weng. Elliptic curves suitable for pairing based cryptography. *Des. Codes Cryptography*, 37:133–141, October 2005. ISSN 0925-1022.

[96] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In Bart Preneel and Stafford E. Tavares, editors, *Selected Areas in Cryptography*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331. Springer, 2005. ISBN 3-540-33108-5.

[97] Ezekiel J. Kachisa, Edward F. Schaefer, and Michael Scott. Constructing brezing-weng pairing-friendly elliptic curves using elements in the cyclotomic field. In *Proceedings of the 2nd international conference on Pairing-Based Cryptography*, Pairing '08, pages 126–135, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-85503-3.

[98] David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *J. Cryptol.*, 23(2):224–280, 2010. ISSN 0933-2790.

[99] Michael Scott and Paulo S. Barreto. Generating more mnt elliptic curves. *Des. Codes Cryptography*, 38:209–217, February 2006. ISSN 0925-1022.

[100] Steven D. Galbraith, James F. McKee, and P. C. Valena. Ordinary abelian varieties having small embedding degree. *Finite Fields and Their Applications*, 13(4):800–814, 2007.

[101] D. Page, N. P. Smart, and F. Vercauteren. A comparison of mnt curves and supersingular curves. *Appl. Algebra Eng., Commun. Comput.*, 17:379–392, October 2006. ISSN 0938-1279.

[102] David Freeman. Constructing pairing-friendly elliptic curves with embedding degree 10. In *10th Workshop on Elliptic Curves in Cryptography (ECC 2006*, pages 452–465. Springer-Verlag, 2006.

[103] Trusted Computing Group. Online at `https://www.`

trustedcomputinggroup.org/, 2011.

[104] L. Chen and T. P. Pedersen. New group signature schemes. In A. De Santis, editor, *Advances in Cryptology- EUROCRYPT'94*, pages 171–181. Springer, Berlin,, 1994.

[105] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '00, pages 255–270, London, UK, 2000. Springer-Verlag. ISBN 3-540-67907-3.

[106] Giuseppe Ateniese, Dawn Song, and Gene Tsudik. Quasi-efficient revocation of group signatures. In *Proceedings of the 6th international conference on Financial cryptography*, FC'02, pages 183–197, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-00646-X.

[107] Dawn Xiaodong Song. Practical forward secure group signature schemes. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, CCS '01, pages 225–234, New York, NY, USA, 2001. ACM. ISBN 1-58113-385-5.

[108] Giuseppe Ateniese and Gene Tsudik. Some open issues and new directions in group signatures. In *Proceedings of the Third International Conference on Financial Cryptography*, FC '99, pages 196–211, London, UK, 1999. Springer-Verlag. ISBN 3-540-66362-2.

[109] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '91, pages 129–140, London, UK, 1992. Springer-Verlag. ISBN 3-540-55188-3.

[110] David Chaum and Hans Van Antwerpen. Undeniable signatures. In *Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology*,

CRYPTO '89, pages 212–216, London, UK, UK, 1990. Springer-Verlag. ISBN 3-540-97317-6.

[111] Lidong Chen and Torben P. Pedersen. Group signatures: Unconditional security for members. In *Technical Report DAIMI PB-481*. Aarhus University, September, 1994.

[112] Lidong Chen and Torben P. Pedersen. On the efficiency of group signatures providing information-theoretic anonymity. In *Proceedings of the 14th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'95, pages 39–49, Berlin, Heidelberg, 1995. Springer-Verlag. ISBN 3-540-59409-4.

[113] Adi Shamir. How to share a secret. *Commun. ACM*, 22:612–613, November 1979. ISSN 0001-0782.

[114] Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In *FOCS*, pages 427–437. IEEE Computer Society, 1987.

[115] Seung Joo Kim, Sung Jun Park, and Dong Ho Won. Convertible group signatures. In *Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology*, pages 311–321, London, UK, 1996. Springer-Verlag. ISBN 3-540-61872-4.

[116] S.J. Park, I.S. Lee, and D.H. Won. A practical group signature. In *Proceedings of the 1995 Japan-Korea Workshop on Information Security and Cryptography*, pages 127–133, 1995.

[117] Markus Michels. Comments on some group signature schemes. Technical report, Department of Computer Science, University of Technology, Chemnitz, Department of Computer Science, University of Technology, Chemnitz, November 1996.

[118] Chae Hoon Lim and Pil Joong Lee. Remarks on convertible group signatures of asiacrypt'96, 1997.

[119] Shahrokh Saeednia. On the security of a convertible group signature scheme. *Inf. Process. Lett.*, 73:93–96, February 2000. ISSN 0020-0190.

[120] Jan Camenisch. *Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem*. PhD thesis, ETH Zurich, 1998. Reprint as vol. 2 of *ETH Series in Information Security and Cryptography*, ISBN 3-89649-286-1, Hartung-Gorre Verlag, Konstanz, 1998.

[121] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber, editors, *STOC*, pages 365–377. ACM, 1982. ISBN 0-89791-070-2.

[122] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[123] Jan Camenisch and Markus Michels. Separability and efficiency for generic group signature schemes. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 413–430, London, UK, 1999. Springer-Verlag. ISBN 3-540-66347-9.

[124] Don Coppersmith. Finding a small root of a bivariate integer equation; factoring with high bits known. In *Proceedings of the 15th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'96, pages 178–189, Berlin, Heidelberg, 1996. Springer-Verlag. ISBN 3-540-61186-X.

[125] Jan Camenisch and Jens Groth. Group signatures: Better efficiency and new theoretical aspects. In Carlo Blundo and Stelvio Cimato, editors, *SCN*, volume 3352 of *Lecture Notes in Computer Science*, pages 120–133. Springer, 2004. ISBN 3-540-24301-1.

[126] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In *Proceedings of the 3rd international conference on Security in communica-*

tion networks, SCN'02, pages 268–289, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-00420-3.

[127] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '02, pages 61–76, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-44050-X.

[128] Henrik Slot Hansen and Kristoffer Kjævik Pagels. Analyse og implementation af fem gruppesignatursystemer, 2006. URL `http://www.daimi.au.dk/~slot/speciale/speciale.pdf`.

[129] Jun Furukawa and Hideki Imai. An efficient group signature scheme from bilinear maps. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E89-A:1328–1338, May 2006. ISSN 0916-8508.

[130] Ross Anderson. Invited Lecture, 4th ACM Computer and Communications Security, 1997.

[131] Mihir Bellare and Sara K. Miner. A forward-secure digital signature scheme. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 431–448, London, UK, 1999. Springer-Verlag. ISBN 3-540-66347-9.

[132] Emmanuel Bresson and Jacques Stern. Efficient revocation in group signatures. In *Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography*, PKC '01, pages 190–206, London, UK, 2001. Springer-Verlag. ISBN 3-540-41658-7.

[133] Ran Canetti and Shafi Goldwasser. An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack. In *Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, EUROCRYPT'99, pages 90–106, Berlin, Heidelberg, 1999. Springer-Verlag. ISBN

3-540-65889-0.

[134] Gene Itkis and Leonid Reyzin. Forward-secure signatures with optimal signing and verifying. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '01, pages 332–354, London, UK, 2001. Springer-Verlag. ISBN 3-540-42456-3.

[135] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In Colin Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer, 2001. ISBN 3-540-42987-5.

[136] David Chaum. Security without identification: transaction systems to make big brother obsolete. *Commun. ACM*, 28:1030–1044, October 1985. ISSN 0001-0782.

[137] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology*, pages 13–25, London, UK, 1998. Springer-Verlag. ISBN 3-540-64892-5.

[138] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *Proceedings of the 16th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'97, pages 256–266, Berlin, Heidelberg, 1997. Springer-Verlag. ISBN 3-540-62975-0.

[139] Lan Nguyen and Reihaneh Safavi-Naini. Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings. In Pil Joong Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 372–386. Springer, 2004. ISBN 3-540-23975-8.

[140] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '91, pages 433–444, London, UK, 1992. Springer-Verlag. ISBN 3-540-55188-3.

[141] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, STOC '90, pages 427–437, New York, NY, USA, 1990. ACM. ISBN 0-89791-361-2.

[142] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 302–321. Springer, 2005. ISBN 3-540-25910-4.

[143] Steven D. Galbraith and Victor Rotger. Easy decision-diffie-hellman groups. *LMS Journal of Computation and Mathematics*, 7:201–218, 2004.

[144] Ernest F. Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In Atluri et al. [176], pages 132–145. ISBN 1-58113-961-6.

[145] *TCG, Specification Version 1.2*. Trusted Computing Group, Incorporated., 2003. `https://www.trustedcomputinggroup.org`.

[146] *ISO/IEC 11889*. Information Technology - Trusted Platform Module, 2009.

[147] He Ge and Stephen R. Tate. A direct anonymous attestation scheme for embedded devices. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 16–30. Springer, 2007. ISBN 978-3-540-71676-1.

[148] Ernie Brickell and Jiangtao Li. Enhanced privacy id from bilinear pairing. *IACR Cryptology ePrint Archive*, 2009:95, 2009.

[149] Liqun Chen. A daa scheme requiring less tpm resources. In Feng Bao, Moti Yung, Dongdai Lin, and Jiwu Jing, editors, *Inscrypt*, volume 6151 of *Lecture Notes in Computer Science*, pages 350–365. Springer, 2009. ISBN 978-3-642-16341-8.

[150] Ernie Brickell and Jiangtao Li. Enhanced privacy id from bilinear pairing for hardware authentication and attestation. In Ahmed K. Elmagarmid and Divyakant

Agrawal, editors, *SocialCom/PASSAT*, pages 768–775. IEEE Computer Society, 2010. ISBN 978-0-7695-4211-9.

[151] Ernie Brickell, Liqun Chen, and Jiangtao Li. Simplified security notions of direct anonymous attestation and a concrete scheme from pairings. *Int. J. Inf. Sec.*, 8(5): 315–330, 2009.

[152] Liqun Chen. A daa scheme using batch proof and verification. In Acquisti et al. [179], pages 166–180. ISBN 978-3-642-13868-3.

[153] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289. Springer, 2002. ISBN 3-540-00420-3.

[154] Liqun Chen and Jiangtao Li. A note on the chen-morrissey-smart daa scheme. *Inf. Process. Lett.*, 110(12-13):485–488, 2010.

[155] Man Ho Au, Willy Susilo, and Yi Mu. Constant-size dynamic *k*-taa. In Roberto De Prisco and Moti Yung, editors, *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 111–125. Springer, 2006. ISBN 3-540-38080-9.

[156] Liqun Chen and Jiangtao Li. Revocation of direct anonymous attestation. In Liqun Chen and Moti Yung, editors, *INTRUST*, volume 6802 of *Lecture Notes in Computer Science*, pages 128–147. Springer, 2010. ISBN 978-3-642-25282-2.

[157] Boston suburb secures metro-scale wireless mesh network with bluesocket. Available at `http://www.tmcnet.com/usubmit/2006/09/27/1936581.htm`, 2006.

[158] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith. Blacklistable anonymous credentials: Blocking misbehaving users without ttps. In *ACM CCS 2007*, pages 72–81. ACM, 2007.

[159] M. Barbaro and T. Jr. Zeller. A face is exposed for AOL searcher no. 4417749. *The New York Times*, 2006. URL `http://www.nytimes.com/2006/08/09/technology/09aol.html`. Accessed 25-February-2010.

[160] Oded Goldreich. *The Foundations of Cryptography — Volume 2, Basic Applications*. Cambridge University Press, May 2004. ISBN 0-521-83084-2.

[161] Sebastian Clauß. A framework for quantification of linkability within a privacy-enhancing identity management system. In Günter Müller, editor, *ETRICS*, volume 3995 of *Lecture Notes in Computer Science*, pages 191–205. Springer, 2006. ISBN 3-540-34640-6.

[162] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 2560 (Proposed Standard), June 1999. URL `http://www.ietf.org/rfc/rfc2560.txt`. Updated by RFC 6277.

[163] S. Santesson and P. Hallam-Baker. Online Certificate Status Protocol Algorithm Agility. RFC 6277 (Proposed Standard), June 2011. URL `http://www.ietf.org/rfc/rfc6277.txt`.

[164] Trusted Computing Group. *TPM Main Specification Level 2 Version 1.2, Revision 103*, July 2007. Available at `http://www.trustedcomputinggroup.org/`.

[165] Ross Anderson. Two remarks on public key cryptology. Technical Report UCAM-CL-TR-549, University of Cambridge Computer Laboratory, United Kingdom, 2002.

[166] Robert P. Gallant, Robert J. Lambert, and Scott A. Vanstone. Faster point multiplication on elliptic curves with efficient endomorphisms. In Kilian [175], pages 190–200. ISBN 3-540-42456-3.

[167] Ahmet Onur Durahim and Erkay Savas. A^2-MAKE: An efficient anonymous and

accountable mutual authentication and key agreement protocol for wmns. *Ad Hoc Networks*, 9(7):1202–1220, 2011.

[168] Michael Scott, Neil Costigan, and Wesam Abdulwahab. Implementing cryptographic pairings on smartcards. In Louis Goubin and Mitsuru Matsui, editors, *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 134–147. Springer, 2006. ISBN 3-540-46559-6.

[169] Piotr Szczechowiak, Anton Kargl, Michael Scott, and Martin Collier. On the application of pairing based cryptography to wireless sensor networks. In David A. Basin, Srdjan Capkun, and Wenke Lee, editors, *WISEC*, pages 1–12. ACM, 2009. ISBN 978-1-60558-460-7.

[170] Standard Specifications for Public Key Cryptography IEEE P1363 / D13. Annex A, Number-Theoretic Background, November 1999. `http://grouper.ieee.org/groups/1363/`.

[171] Sanjit Chatterjee, Darrel Hankerson, Edward Knapp, and Alfred Menezes. Comparing two pairing-based aggregate signature schemes. *Des. Codes Cryptography*, 55:141–167, May 2010. ISSN 0925-1022.

[172] Michael Scott. *MIRACL—A Multiprecision Integer and Rational Arithmetic C/C++ Library*. Shamus Software Ltd, Dublin, Ireland, 2009. Available at `http://ftp.computing.dcu.ie/pub/crypto/miracl.zip`.

[173] Benoît Chevallier-Mames, Jean-Sébastien Coron, Noel McCullagh, David Naccache, and Michael Scott. Secure delegation of elliptic-curve pairing. In Gollmann et al. [180], pages 24–35. ISBN 978-3-642-12509-6.

[174] The ns-3 network simulator. Available at `http://www.nsnam.org`.

[175] Joe Kilian, editor. *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, 2001.

Springer. ISBN 3-540-42456-3.

[176] Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick Drew McDaniel, editors. *Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washington, DC, USA, October 25-29, 2004*, 2004. ACM. ISBN 1-58113-961-6.

[177] Matthew K. Franklin, editor. *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, 2004. Springer. ISBN 3-540-22668-0.

[178] Steven D. Galbraith and Kenneth G. Paterson, editors. *Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1-3, 2008. Proceedings*, volume 5209 of *Lecture Notes in Computer Science*, 2008. Springer. ISBN 978-3-540-85503-3.

[179] Alessandro Acquisti, Sean W. Smith, and Ahmad-Reza Sadeghi, editors. *Trust and Trustworthy Computing, Third International Conference, TRUST 2010, Berlin, Germany, June 21-23, 2010. Proceedings*, volume 6101 of *Lecture Notes in Computer Science*, 2010. Springer. ISBN 978-3-642-13868-3.

[180] Dieter Gollmann, Jean-Louis Lanet, and Julien Iguchi-Cartigny, editors. *Smart Card Research and Advanced Application, 9th IFIP WG 8.8/11.2 International Conference, CARDIS 2010, Passau, Germany, April 14-16, 2010. Proceedings*, volume 6035 of *Lecture Notes in Computer Science*, 2010. Springer. ISBN 978-3-642-12509-6.