Unconditionally Secure Digital Signature

Francesco Auriemma

November 2003

Coordinatore: Prof. Alfredo De Santis
Relatori: Prof. Alfredo De Santis
Dr. Paolo D’Arco
Abstract

Digital signatures are rapidly becoming ubiquitous in many aspects of electronic life. They are used to obtain security services such as authentication, data integrity and non-repudiation. A relatively new concept is group signature that allows any member of a potentially large group to sign on behalf of the group.

A potentially serious problem with current digital signature schemes is that, the underlying hard problems from number theory on which these schemes are based on, may be solved by innovative techniques or by a new generation of computing devices such as quantum computers. Therefore, while these signature schemes represent an efficient solution to the short term integrity (unforgeability and non-repudiation) of digital data, they provide no confidence on the long term (say 20 years) integrity of data signed by these schemes. In the past some researchers have focused their attention on signature schemes whose security does not rely on any unproven assumption.

In this thesis we establish two models for unconditionally secure digital signature, which can be used to generate both user and group signatures. The given models are different in the use of a Trusted Authority. In the first model the Trusted Authority is used in the Key Distribution phase and, later on, in order to solve disputes, while, in the second, he only distributes the secret information. For both models we give two constructions referred to as a symmetric and an asymmetric constructions, respectively.
Acknowledgements

I would express my gratitude to my advisor Alfredo De Santis and to Paolo D’Arco for their help and friendship. I am also grateful to all others professors and Ph.D. students at the Dipartimento di Informatica ed Applicazioni. All of them made my graduate experience enjoyable and educational.
Contents

Abstract v

Acknowledgements vi

List of Tables ix

1 Digital Signatures and Group Signatures 1
   1.1 Signatures schemes ........................................ 1
   1.1.1 Some important signature schemes ...................... 2
   1.2 Group Signatures ........................................ 4
      1.2.1 Basic concepts and security definition .............. 5
      1.2.2 Some important Group Signatures schemes .......... 8
   1.3 Why Unconditionally Secure Signatures? .................. 10

2 Multireceiver authentication codes (MRA): Model and Bounds 13
   2.1 What are Multireceiver Authentication codes? ......... 13
   2.2 Preliminaries ........................................... 14
   2.3 Model and bounds ....................................... 16

3 Some Constructions of MRA-code 23
   3.1 The DFY polynomial construction ...................... 23
   3.2 Generalizations ......................................... 24
      3.2.1 MRA-code for multiple message transmissions .... 24
      3.2.2 MRA-codes with dynamic sender .................... 29
4 Unconditionally secure signatures

4.1 Towards Unconditionally secure signatures: Transferability

4.1.1 An attack on DMRA of Section 3.2.2

4.1.2 A Method to fix the problem

4.2 Models of unconditionally secure signature

4.2.1 Attacks against the schemes

4.2.2 Two constructions of unconditionally secure signature schemes

4.2.3 Multiple messages signature scheme

5 Unconditionally Secure Group Signatures

5.1 What are Unconditionally Secure Group Signatures?

5.2 Unconditionally Secure Group Signature opened by TA

5.2.1 The model

5.2.2 Properties and attacks

5.2.3 A symmetric construction

5.2.4 Reducing memory storage of the symmetric construction

5.2.5 An asymmetric construction

5.2.6 Reducing memory storage of the asymmetric construction

5.2.7 Consideration on the security parameter $q'$

5.3 Unconditionally Secure Group Signature opened by the signer

5.3.1 The model

5.3.2 Properties and attacks

5.3.3 A symmetric construction

5.3.4 An asymmetric construction

6 Conclusions

References

A Proof of Theorem 2.4
List of Tables

1.1 Experiment for Full-anonymity .............................. 7
1.2 Experiment for Full-traceability ............................. 8
1.3 Factoring records since 1970 ............................... 11
Chapter 1

Digital Signatures and Group Signatures

1.1 Signatures schemes

In a signature scheme, each user publishes a public key and keeps for himself a secret key. A user’s signature on a message $m$ is a value which depends on $m$ and on user’s public and secret keys in such a way that anyone can check validity just by using the public key. However, it results hard to forge a user’s signature without knowing his secret key. A signature scheme is defined by:

- the *key generation algorithm* $\mathcal{G}$ which, on input $1^k$, where $k$ is the security parameter, produces a pair $(K_p, K_s)$ of matching public and secret keys. It is clear that $\mathcal{G}$ must be a probabilistic algorithm.

- the *signing algorithm* $\Sigma$ which, given a message $m$ and a pair of matching public and secret keys $(K_p, K_s)$, produces a signature. The signing algorithm might be probabilistic, and in some schemes it might receive other inputs as well.

- the *verification algorithm* $\mathcal{V}$ which, given a signature $\sigma$, a message $m$ and a public key $K_p$, tests whether $\sigma$ is a valid signature of $m$ with respect to $K_p$. In general, the verification algorithm need not be probabilistic.

Usually a signature is not applied directly on a message $m$ but on the hash value $\mathcal{H}(m)$, generated by using a hash function $\mathcal{H}$. 
1.1.1 Some important signature schemes

Digital signatures are an important technology for ensuring unforgeability and non-repudiation of digital data. Currently, signature schemes rely on the presumed computational difficulty of solving certain number theoretic problems, such as factoring large composites or computing discrete logarithms in a large finite field. In the following we will describe briefly some important signature schemes:

- **RSA Signatures**
  
  Let $n = pq$, with $p$ and $q$ primes. Let $(e, n)$ be Alice’s public key and $(d, n)$ the corresponding secret key such that $ed \equiv 1 \mod \varphi(n)$ with $\varphi(n) = (p - 1)(q - 1)$. To sign a message $m$, it is sufficient to compute $C = H(m)^d \mod n$ where $H(\cdot)$ is a hash function defined as $H : \{0, 1\}^* \rightarrow \mathbb{Z}_n$. The signature is accepted only if $C^e \mod n$ matches $H(m)$.

- **Gennaro-Halevi-Rabin Signatures**
  
  Let $n$ be the product of two ”safe” primes $p = 2p' + 1$ and $q = 2q' + 1$. Alice’s certified public-key is $(n, s)$, where $s$ is randomly chosen in $\mathbb{Z}_n^*$. In order to sign a message $m$, Alice computes $e = H(m)$ and $\sigma = s^e \mod n$. To verify the signature, Bob computes $e = H(m)$ and checks whether $\sigma^e \equiv s \mod n$. The Gennaro-Halevi-Rabin signature scheme [29] has been proven resistant against adaptive chosen message attacks, in the random oracle model, under the strong RSA assumption\(^1\). The authors in [29] provide other constructions eliminating the need for the random oracle model.

- **Cramer-Shoup Signatures**
  
  Let $j$ and $z$ be two security parameters such that $j + 1 < z$. Let $n = pq$, where $p$ and $q$ are $z$-bit safe primes, i.e., $p = 2p' + 1$ and $q = 2q' + 1$ where both $p'$ and $q'$ primes. Alice’s public key is $(n, b, x, e')$, where $b, x$ are randomly chosen in the subgroup of quadratic residues modulo $n$, $Q_n$ and $e'$ is a random $(j + 1)$-bit prime. To generate a Cramer-Shoup signature [21] on a message $m$, Alice randomly selects a $(j + 1)$-bit prime $e \neq e'$ and $u' \in Q_n$. Then, she computes $x' = (u')^{e'} b^{-H(m)}$ and finally $u = (xb^{H(x')})^{\frac{1}{2}}$ where $H(\cdot)$ is a hash function defined as $H : \{0, 1\}^* \rightarrow \{0, 1\}^j$. The

\(^1\)The strong RSA assumption states that given a random $z \in \mathbb{Z}_n^*$ for a certain RSA modulus $n$, finding a pair $(u, e) \in \mathbb{Z}_n^* \times \mathbb{Z}$ such that $e > 1$ and $u^e = z$ is hard to solve
resulting signature on $m$ is $(e, u, u')$. To verify it, Bob checks that $e$ is a $(j + 1)$-bit number different from $e'$ and computes $x' = u^{e'} b^{-H(x')}$. Then, Bob checks whether $x = u^e b^{-H(x')}$. The Cramer-Shoup signatures is quite efficient and it is provable secure against adaptive chosen message attacks under the strong RSA assumption.

- **Guillou-Quisquater Signatures**

A trusted third party generates common parameters $v, n = pq$, and for each user, a secret key $B$ and an ID-based public-key $J$ such that $B^v J \equiv 1 \mod n$. Only the trusted third party knows the order of $\mathbb{Z}_n^*$. The Guillou-Quisquater signature [31] on a message $m$ is computed as follows: randomly choose $r \in \mathbb{Z}_n^*$ and compute $T = r^v \mod n$, $d = H(m || T)$ and $D = rB^d \mod n$, where $H(\cdot)$ is a suitable hash function. The resulting signature is the pair $(d, D)$. The trusted third party, which has generated the system parameters, may cease to exist after the initialization phase. To verify the signature, it is sufficient to check whether $d = H(m || D^v J^d)$. Indeed notice that $D^v J^d \equiv (rB^d)^v J^d \equiv r^v (B^v J)^d \equiv r^v \mod n$.

- **Schnorr and Poupard-Stern Signatures**

Let $\alpha$ be a generator of the unique cyclic subgroup of prime order $q$ in $\mathbb{Z}_p^*$, where $p$ is some large prime number such that $q$ divides $p - 1$. Alice (the signer) selects the private key $1 \leq a \leq q$, computes $y = \alpha^a \mod p$ and publishes $(p, q, \alpha, y)$. To generate a Schnorr signature on a message $m$, Alice selects a random secret integer $k$, with $1 \leq k \leq q - 1$, and computes $r = \alpha^k \mod p$, $e = H(m || r)$ where $H(\cdot)$ is a hash function defined as $H : \{0, 1\}^* \to \mathbb{Z}_q$, and $s = ae + k \mod q$. Alice’s signature on message $m$ is the pair $(s, e)$. Bob verifies the signature by checking whether $e = e'$, where $e' = H(m || \alpha^s y^{-e})$. Like to previous scheme is the Poupard-Stern signature scheme [48] that is the Schnorr signature modulo a composite. The Poupard-Stern signature has formally been proven resistant against chosen message attacks in the random oracle model, under the discrete logarithm assumption.

- **ElGamal Signatures**

Let $\alpha$ be a generator of the unique cyclic subgroup of prime order $q$ in $\mathbb{Z}_p^*$, where $p$ is some large prime number such that $q$ divides $p - 1$. Alice (the signer) selects the private key $1 \leq a \leq q$, computes $y = \alpha^a \mod p$ and publishes $(p, q, \alpha, y)$. To
Chapter 1. Digital Signatures and Group Signatures

generate an ElGamal signature on a message \( m \), Alice selects a random \( k \in [p - 2] \) and computes \( r = \alpha^k \mod p \) and \( s = ar + k\mathcal{H}(m) \mod q \). The signature is the pair \((r, s)\). To verify it, Bob checks that \( 1 \leq r \leq p - 1 \) and tests whether \( \alpha^r = y^r \cdot \mathcal{H}(m) \).

[28]

1.2 Group Signatures

Group Signature is a relatively new concept introduced by Chaum and Van Heyst in 1991 [16]. A group signature - similarly to its traditional counterpart - allows a signer to demonstrate knowledge of a secret with respect to a specific document. It is also publicly verifiable: anyone in possession of a group public key can validate a group signature. However, group signatures are anonymous in that, no one, with the sole exception of a designated group manager, can discover the identity of the signer. Furthermore, group signatures are unlinkable, i.e., it is computationally hard to establish whether or not multiple signatures are produced by the same group member. In exceptional cases (such as a legal dispute) any group signature can be "opened" by the group manager to reveal unambiguously the identity of the actual signer. At the same time, no one (including the group manager) can misattribute a valid group signature.

The salient features of group signatures make them attractive for many specialized applications such as voting and bidding. They can, for example, be used in vitiation to submit tenders [18]. All companies submitting a tender form a group and each company signs its tender anonymously using the group signature. Once the preferred tender is selected, the winner can be traced while the other bidders remain anonymous. More generally, group signatures can be used to conceal organizational structures, e.g., when a company or a government agency issues a signed statement. Group signature can also be integrated with an electronic cash system whereby several banks can securely distribute anonymous and untraceable e-cash. This permits concealing of the cash-issuing banks' identities [43].

\[\text{2This is a modified version of the ElGamal signature scheme since we are working in a prime-order subgroup of \( Z_p \) whereas the original scheme works directly in \( Z_p \).}\]
1.2.1 Basic concepts and security definition

Before giving a formal definition and security notions of group signature schemes we describe notations and terminology. If \( x \) is a string, then \( |x| \) denotes its length, while if \( S \) is a set then \( |S| \) denotes its size. The empty string is denoted by \( \epsilon \). If \( k \in \mathcal{N} \) then \( 1^k \) denotes the string of \( k \) ones. If \( n \) is an integer then \( [n] = \{1, \ldots, n\} \). If \( A \) is a randomized algorithm then \( z \leftarrow A(x, y, \cdots) \) denotes the operation of running \( A \) on input \( x, y, \cdots \) and letting \( z \) be the output. If \( A \) is a randomized algorithm then \( [A(x, y, \cdots)] \) denotes the set of all points having positive probability of being output by \( A \) on input \( x, y, \cdots \). We say that a function \( f : \mathcal{N} \to \mathcal{N} \) is nice if it is polynomially bounded and computable in \( \text{poly}(k) \) time. A function \( g : \mathcal{N} \to \mathcal{R} \) is called negligible if it approaches zero faster than the reciprocal of any polynomial. That is, for any \( c \in \mathcal{N} \) there is an integer \( n_c \) such that \( g(n) \leq n^{-c} \) for all \( n \geq n_c \).

Group signature schemes are defined as follows

**Definition 1.1** [5] A group signature scheme \( GS = (Gk_g, GSig, GVf, Open) \) consists of four polynomial-time algorithms:

- The randomized group key generation algorithm \( (\text{SETUP}) \) takes input \( 1^k, 1^n \), where \( k \in \mathcal{N} \) is the security parameter and \( n \in \mathcal{N} \) is the group size, and returns a tuple \( (gpk, gmsk, gsk) \), where \( gpk \) is the group public key, \( gmsk \) is the group manager’s secret key, and \( gsk \) is an \( n \)-vector of keys with \( gsk[i] \) being a secret signing key for player \( i \in [n] \).

- The randomized group signing algorithm \( GSig \) takes as input the secret signing key \( gsk[i] \) and a message \( m \) to return a signature of \( m \) under \( gsk[i] \) (\( i \in [n] \)).

- The deterministic group signature verification algorithm \( GVf \) takes as input the group public key \( gpk \), a message \( m \) and a candidate signature \( \sigma \) for \( m \) to return either 1 or 0.

- The deterministic opening algorithm \( Open \) takes as input the group manager secret key \( gmsk \), a message \( m \), and a signature \( \sigma \) of \( m \) to return an identity \( i \) or the symbol \( \bot \).
Chapter 1. Digital Signatures and Group Signatures

For simplicity we are assigning to group members consecutive integer identities 1, 2, \ldots, n. We say that \( \sigma \) is a true signature of \( m \) if there exist \( i \in [n] \) such that \( \sigma \in GSig(gsk[i], m) \). We say that \( \sigma \) is a valid signature of \( m \) with respect to gpk if \( GVf(gpk, m, \sigma) = 1 \).

The properties that a group signature scheme must satisfy are:

- **Correctness:** The scheme must satisfy the following correctness requirement: For all \( k, n \in \mathbb{N} \), all \( (gpk, gmsk, gsk) \in [GKg(1^k, 1^n)] \), all \( i \in [n] \) and all \( m \in \{0, 1\}^* \)

  \[
  GVf(gpk, m, GSig(gsk[i], m)) = 1 \quad \text{and} \quad \text{Open}(gmsk, m, GSig(gsk[i], m)) = 1.
  \]

  The first says that true signatures are always valid. The second asks that the opening algorithm correctly recovers the identity of the signer from a true signature.

- **Compactness:** In practice it is preferable that sizes of keys and signatures in a group signature scheme do not grow proportionally to the number of members \( n \). We call a group signature scheme \( GS = (GKg, GSig, GVf, Open) \) compact if there exist polynomials \( p_1(\cdot, \cdot) \) and \( p_2(\cdot, \cdot, \cdot) \) such that

  \[
  |gpk|, |gmsk|, |gsk[i]| \leq p_1(k, \log(n)) \ \text{and} \ |\sigma| \leq p_2(k, \log(n, |m|))
  \]

  for all \( k, n \in \mathbb{N} \), all \( (gpk, gmsk, gsk) \in [GKg(1^k, 1^n)] \), all \( i \in [n] \), all \( m \in \{0, 1\}^* \) and all \( \sigma \in [GSig(gsk[i], m)] \).

- **Full-Anonymity:** Informally, anonymity requires that an adversary who does not possess the group manager’s secret key finds it hard to recover the identity of the signer from his signature. To define the full-anonymity we superimpose an adversary with strong attack capabilities. In particular, to capture the possibility of an adversary colluding with group members we give it the secret keys of all group members. To capture the possibility of seeing the results of previous openings by the group manager, we give it access to an opening oracle \( \text{Open}(gmsk, \cdot, \cdot) \). Let us now proceed to the formalization. To any group signature scheme \( GS = (GKg, GSig, GVf, Open) \), an adversary \( A \) and bit \( b \) we associate the experiment in Table 1.1. Here, \( A \) is an adversary that functions in two stages, a choose stage and a guess stage. In the choose stage, \( A \) takes as input the group members secret key, gsk, together with the group public key gpk. During this stage, it can also query the opening oracle \( \text{Open}(gmsk, \cdot) \) on group signature of his choice and it is required that at the end of
Chapter 1. Digital Signatures and Group Signatures

Table 1.1: Experiment for Full-anonymity

stage $A$ outputs two valid identities $1 \leq i_0, i_1 \leq n$, and a message $m$. The adversary also outputs some state information to be used in the second stage of the attack. In the second stage, the adversary is given the state information, and a signature on $m$ produced using the secret key of one of the two users $i_0, i_1$, chosen at random. The goal is to guess which of the two secret keys was used. The adversary can still query the opening oracle, but not on the challenge signature. We denote by

$$Adv_{\text{anon}}^{\text{GS,A}}(k, n) = Pr[Exp_{\text{anon}}^{-1}(k, n) = 1] - Pr[Exp_{\text{anon}}^{0}(k, n) = 1]$$

the advantage of adversary $A$ in breaking the full-anonymity of $GS$. We say that a group signature scheme is full-anonymous if for any polynomial-time adversary $A$, the two-argument function $Adv_{\text{anon}}^{\text{GS,A}}(\cdot, \cdot)$ is negligible.

• Full-Traceability: In case of misuse, signer anonymity can be revoked by the group manager. In order for this to be an effective deterrence mechanism, we require that no colluding set $S$ of group members can create signatures that cannot be opened, or signatures that cannot be traced back to some member of the coalition. We formally define full-traceability using the experiment of Table 1.2. Here, adversary $A$ runs in two stages, a choose stage and a guess stage. On input the group public key gpk and the secret of group manager, gmsk, the adversary starts its attack by adaptively corrupting a set $C$ of group members. The identities of the group members that are corrupted and their number is entirely up to it. At the end of the choose stage the set $C$ contains the identities of the corrupted members. In the guess stage, the adversary attempts to produce a forgery $(m, \sigma)$, and we say it wins if $\sigma$ is a valid group signature of $m$, but the opening algorithm return $\perp$ or some valid user identity
Chapter 1. Digital Signatures and Group Signatures

Experiment  $\text{Exp}^{\text{trace}}_{GS,A}(k, n)$

$\text{Pr} \left[ \text{Exp}^{\text{trace}}_{GS,A}(k, n) = 1 \right]$

\[
\begin{align*}
(gpk, gmsk, gsk) & \xleftarrow{\$} G K g(1^k, 1^n) \\
St & \xleftarrow{\$} (gmsk, gpk); C \leftarrow \emptyset; K \leftarrow \epsilon; \text{Count} \leftarrow \text{true} \\
\text{while} (\text{Count} = \text{true}) \text{ do} \\
& (\text{Count}, St, j) \xleftarrow{\$} A^{GSig(gsk[\cdot])}(\text{choose}, St, K) \\
& \text{if} \text{ Count} = \text{true} \text{ then } C \leftarrow \cup \{j\}; K \leftarrow gsk[j] \text{ EndIf} \\
\text{Endwhile} \\
(m, \sigma) & \xleftarrow{\$} A^{GSig(gsk[\cdot])}(\text{guess}, St) \\
\text{If} G V f(gpk, m, \sigma) = 0 \text{ then return 0; If Open}(gmsk, m, \sigma) = \perp \text{ return 1} \\
\text{If there exists } i \in [n] \text{ such that the following are true then return 1} \\
\text{else return 0:} \\
& 1. \text{Open}(gmsk, m, \sigma) = i \\
& 2. i \notin C \\
& 3. i, m \text{ was not queried by } A \text{ to its oracle}
\end{align*}
\]

Table 1.2: Experiment for Full-traceability

\(\Sigma\) such that \(i \notin C\). Otherwise, the experiment returns 0. We define the advantage of adversary \(A\) in defeating full-traceability of the group signature scheme \(GS\) by:

\[
\text{Adv}^{\text{trace}}_{GS,A}(k, n) = \text{Pr}[\text{Exp}^{\text{trace}}_{GS,A}(k, n) = 1],
\]

and say that \(GS\) is full-traceable if for any polynomial-time adversary \(A\), the two-argument function \(\text{Adv}^{\text{trace}}_{GS,A}(\cdot, \cdot)\) is negligible.

1.2.2 Some important Group Signatures schemes

As we have explained before, the concept of group signatures was introduced and realized by Chaum and Van Heist in 1991 [16]. They proposed four schemes of which three provide computational anonymity whereas the fourth provides information-theoretic anonymity. Some of the schemes do not allow a group manager to add group members after the initial setup. Others require the group manager to contact each member in order to open a signature, i.e., to reveal the identity of the signer.
A number of improvements and enhancements followed the initial work. Some notable results are due to Chen/Pedersen [19], Camenisch [10], Petersen [47], Camenisch/Standler [11] and Ateniese et al. [2]. In [19], two schemes were proposed providing information theoretic and computational anonymity, respectively. Unfortunately, the proposed scheme allows the group manager to misattribute a signature, i.e., falsely accuse a group member of having signed a message.

In [10] an efficient group signature scheme was presented, providing computational anonymity, ability to add (or remove) group members after the initial setup, and the possibility of being generalized by allowing authorized set of group members to sign collectively (appearing to verifiers as a single signer) on behalf of the group. This scheme can be extended to allow the functionality of the group manager to be shared among several entities. The drawbacks include the size of the public key and the signature size (both proportional to the group size).

As also noted in [11], many previous results exhibit some or all of the following shortcomings:

1. The size of the group public key depends on the size of the group.
2. The length of a group signature depends on the size of the group.
3. New member addition either requires restarting the entire system or involves reissuing all members’ keys and changing the group public key.
4. Exclusion/revocation of group members requires re-issuing all members’ keys and changing the group public key.

The results presented in [11] address the first three of the above issues, albeit, at significant cost. The basic scheme (referred hereafter as CS97), while quite elegant, involves costly computation and reliance on new problems conjectured to be computationally difficult to solve. (One such problem is the difficulty of computing double discrete logarithms in finite groups. Another - the difficulty of computing roots of discrete logarithms.) Nonetheless, CS97 is, in principle, simple and appealing in that the group public key and the group signature are both of constant size.

In [2] Ateniese et al. present a group signature / identity escrow scheme that is provably secure. In particular, the escrow identity scheme is provably coalition resistant under
the strong RSA assumption. Other security properties hold under the decisional Diffie-Hellman or the discrete logarithm assumption. The group signature scheme is obtained from the identity escrow scheme using the Fiat-Shamir heuristic, hence it is secure in the random oracle model.

1.3 Why Unconditionally Secure Signatures?

Digital signature schemes based on number theoretic problems are prevalent methods used in providing data integrity. These schemes rely for their security on the presumed computational difficulty of solving certain number theoretic problems, such as factoring large composites or computing discrete logarithms in large finite fields. Progress in computers as well as further refinements of various algorithms have made it possible to solve the aforementioned number theoretic problems for larger sizes of the input. This can be considered a great problem because while some data may only require the assurance of integrity for a relatively short period of time (say up to 5 years), same other important data, such as court records and speeches by a parliamentarian, require the assurance of integrity for a long period of time (say up to 50 years).

As an example, in August 1999, a team of researchers from around the world succeeded in cracking an 512-bit RSA composite by using the Number Field Sieve \cite{13} over the Internet. The amount of CPU time spent to factor RSA-155 was about 8400 MIPS years\textsuperscript{3}, which is about four times that used for the factorization of RSA-140. Based on the heuristic complexity formula \cite{41} for factoring large $N$ by NFS:

$$
\exp \left( (1.93 + o(1)) (\log N)^{\frac{1}{3}} (\log \log N)^{\frac{2}{3}} \right)
$$

\text{(1.1)}

one would expect an increment in the computing time by a factor of about seven. This speed-up has been made possible by algorithmic improvements, mainly in the polynomial generation step and to a lesser extent in the filter step of NFS.

The complete project for factoring RSA-155 took seven calendar months. The polynomial generation step took about one month on several fast workstations. The most time-consuming step, the sieving, was done on about 300 fast PCs and workstations spread over

\textsuperscript{3}One MIPS year is the equivalent of a computation during one full year at a sustained speed of one Million Instructions Per Second.
twelve "sites" in six countries. This step took 3.7 calendar months, in which, summed over all these 300 computers, a total of 35.7 years of CPU-time was consumed. Filtering the relations and building and reducing the matrix corresponding to these relations took one calendar month and was carried out on an SGI Origin 2000 computer. The block Lanczos step to find dependencies in this matrix took about ten calendar days on one CPU of a Cray C916 supercomputer. The final square root step took about two calendar time days on a SGI Origin 2000 computer.

In order to attempt an extrapolation, we give in Table 1.3 the factoring records starting with the landmark factorization in 1970 by Morrison and Brillhart of $F_7 = 2^{128} + 1$ with the help of the new Continued Fraction (CF) method.

<table>
<thead>
<tr>
<th>number of decimals</th>
<th>date or year</th>
<th>algorithm</th>
<th>effort (MIPS years)</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>Sept 13, 1970</td>
<td>CF</td>
<td></td>
<td>[44]</td>
</tr>
<tr>
<td>50</td>
<td>1983</td>
<td>CF</td>
<td></td>
<td>[9]</td>
</tr>
<tr>
<td>55-71</td>
<td>1983-1984</td>
<td>QS</td>
<td></td>
<td>[22]</td>
</tr>
<tr>
<td>45-81</td>
<td>1986</td>
<td>QS</td>
<td></td>
<td>[62]</td>
</tr>
<tr>
<td>78-90</td>
<td>1987-1988</td>
<td>QS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>87-92</td>
<td>1988</td>
<td>QS</td>
<td></td>
<td>[50]</td>
</tr>
<tr>
<td>93-102</td>
<td>1989</td>
<td>QS</td>
<td></td>
<td>[39]</td>
</tr>
<tr>
<td>107-116</td>
<td>1990</td>
<td>QS</td>
<td></td>
<td>[40]</td>
</tr>
<tr>
<td>RSA-100</td>
<td>Apr 1991</td>
<td>QS</td>
<td>7</td>
<td>[52]</td>
</tr>
<tr>
<td>RSA-110</td>
<td>Apr 1992</td>
<td>QS</td>
<td>75</td>
<td>[26]</td>
</tr>
<tr>
<td>RSA-120</td>
<td>Jun 1993</td>
<td>QS</td>
<td>835</td>
<td>[23]</td>
</tr>
<tr>
<td>RSA-129</td>
<td>Apr 1994</td>
<td>QS</td>
<td>5000</td>
<td>[3]</td>
</tr>
<tr>
<td>RSA-130</td>
<td>Apr 1996</td>
<td>NFS</td>
<td>1000</td>
<td>[20]</td>
</tr>
<tr>
<td>RSA-140</td>
<td>Feb 1999</td>
<td>NFS</td>
<td>2000</td>
<td>[12]</td>
</tr>
</tbody>
</table>

**Table 1.3:** Factoring records since 1970

Let $D$ be the number of decimal digits in the largest "general" number factored by a given date. From the complexity formula for NFS 1.1, assuming Moore’s law (computing
power doubles every 18 months), Brent expect $D^{\frac{1}{3}}$ to be roughly a linear function of the calendar year $Y$. From the data in Table 1.3 he derives the linear formula

$$Y = 13.24D^{\frac{1}{3}} + 1928.6.$$  

According to this formula, a general 768-bit number ($D = 231$) will be factored by the year 2010, and a general 1024-bit number ($D = 309$) by the year 2018.

The above analysis motivates research in studying signatures and group signatures schemes, whose security does not rely on computational assumptions on the power of the adversary. This thesis goes in this direction.
Chapter 2

Multireceiver authentication codes (MRA): Model and Bounds

2.1 What are Multireceiver Authentication codes?

Multireceiver authentication codes (MRA-codes) were introduced by Desmedt, Frankel and Yung (DFY) [25] as an extension of Simmons’ model of unconditionally secure authentication [58]. In a MRA-code, a sender wants to authenticate a message for a group of receivers in such a way that each receiver can verify the authenticity of the received message. The receivers are not trusted and may try to construct fraudulent messages on behalf of the transmitter. If the fraudulent message is acceptable by one receiver, then the attackers have succeeded. This is a useful extension of traditional authentication codes and has numerous applications. For examples a director who wishes to give instructions to employees in an organization in such a way that each employee is able to verify authenticity of the message. Proving such a service using digital signatures implies that security is based on unproven assumptions and the attackers have finite amount of computational resources.

A multireceiver A-code can be trivially constructed using traditional A-codes: the sender shares a common key with each receiver and to send an authenticated message, constructs $n$ codewords, one for each receiver, concatenates them and broadcasts the result. Now each receiver can verify its own codeword and authenticate the message. In this construction collaboration of even $n - 1$ receivers does not enable them to construct a message that is acceptable by the $n^{th}$ receiver simply because the $n$ codewords are independently constructed. However, if we assume that the size of the malicious groups
cannot be too large, for example the biggest number of collaborators is \( \omega - 1 \) (\( \omega < n \)), then we can expect to save on the size of the key and the length of the codeword because codewords can have dependencies. This is the basic motivation for studying MRA-codes that are more efficient than the trivial one described above. DFY description of MRA-code is basically an operational description of the system, that is the way the system works. Kurosawa and Obana (KO) [38] studied \((\omega, n)\) MRA-code, again using the operational description of these codes, derived combinatorial lower bounds on the probability of success in impersonation and substitution attacks, and characterized Cartesian MRA-codes that satisfy the bound with equality. They showed that DFY polynomial construction is in fact an optimal (smallest sizes of transmitter and receiver keys) construction.

### 2.2 Preliminaries

In [58] Simmons’s model of unconditionally secure authentication there are three participants: a transmitter (sender), a receiver and an opponent. The transmitter and the receiver share a secret key and are both assumed honest. The message is sent over a public channel which is subject to active attacks. Transmitter and receiver use an authentication code which is a set of authentication functions \( f \), indexed by a key belonging to a set \( E \). To authenticate a message called a source state and denoted by \( s \in S \), using a key \( e \), the transmitter forms a codeword \( f(e, s) \) from a set \( M \) and sends it to the receiver who can verify its authenticity using his knowledge of the key.

**Definition 2.1** An authentication code \( C \) is a 4-tuple \((S, M, E, f)\), where \( f \) is a mapping from \( S \times E \) to \( M \),

\[
f : S \times E \to M
\]

such that \( f(s, e) = m \) and \( f(s', e) = m \) imply \( s = s' \).

In a systematic Cartesian A-code the codeword corresponding to a source state \( s \) using \( e \in E \) is the concatenation of \( s \) and an authentication tag \( t \in \Gamma \): that is \( m = (s, t) \). The receiver will detect a fraudulent codeword \( (s, t) \) if the tag that he calculates for \( s \) using his secret key \( e \) is different from the received tag \( t \).

The opponent can perform an impersonation, or a substitution attack by constructing a fraudulent codeword, and he succeeds if the codeword is accepted by the receiver. In
Chapter 2. Multireceiver authentication codes (MRA): Model and Bounds

impersonation the attacker has not seen any previous communication, while in substitution he has seen one transmitted codeword. A code provides perfect protection against impersonation if enemy’s best strategy is to randomly guess a codeword. In the case of Cartesian A-codes, enemy’s probability of success is \( P_I = \frac{1}{|\Gamma|} \). Perfect protection for substitution is defined in a similar way and requires enemy’s best strategy to be randomly selecting one of the remaining codewords such that the source state is different from the observed one. For Cartesian A-code the probability of success of the intruder is \( P_S = \frac{1}{|\Gamma|} \).

An extension of this model, proposed by Desmedt, Frankel and Yung (DFY) [25], is when there are multiple receivers. The system works as follows. First the Key Distribution Center (KDC) distributes secret keys to the transmitter and each receiver. Then, the transmitter broadcasts a message to all the receivers who can individually verify authenticity of the message using their secret key information. There are malicious groups of receivers who use their secret keys and all the previous communications in the system to construct fraudulent messages. They succeed in their attack if a receiver accepts the message as being authentic.

The formalization of \((\omega,n)\) MRA-codes, given in [55], is as follows. Let \( E_1, E_2, \ldots, E_n \) denote the set of decoding rules of receivers \( R_1, \ldots, R_n \), and let \( S \) and \( M \) denote the set of source states and senders codewords, respectively. We will also use \( X \) to denote a random variable defined on a set \( X \).

**Definition 2.2 ([38])** We say that \((S,M,E_1,\ldots,E_n)\) is a \((\omega,n)\) multireceiver A-code if for \( \forall(E_{i_1},\ldots,E_{i_\omega}) \) and \( \forall(e_1,\ldots,e_\omega) \),

\[
P(E_{i_\omega} = e_\omega | E_{i_1} = e_1, \ldots, E_{i_{\omega-1}} = e_{\omega-1}) = P(E_{i_\omega} = e_\omega).
\]

Probability of success in impersonation and substitution attacks, \( P_I \) and \( P_S \), for \((\omega,n)\) MRA-codes are defined as

\[
P_I = \max_{E_{i_\omega}} \max_m P(R_{i_\omega} \text{ accepts } m)
\]

\[
P_S = \sum_m P(m) \max_{E_{i_\omega}} \max_{m'} P(R_{i_\omega} \text{ accepts } m' \text{ } | \text{ } R_{i_\omega} \text{ accepts } m)
\]

where the maximum is taken over \( m' \) such that the source state of \( m' \) is different from that of \( m \). With these definitions, they derived the following bounds. Assume \( \ell = |M|/|S| \).
**Theorem 2.1** ([38]) In a \((\omega, n)\) MRA-code, \(P_I \geq 1/\sqrt{\ell}\). Equality holds if and only if \(P(R_{i_1}, \ldots, R_{i_\omega} \text{ accepts } m) = 1/\ell\) and \(P(R_j \text{ accepts } m) = 1/1/\sqrt{\ell}\) for any \(m\) and any \(R_j\).

**Theorem 2.2** ([38]) In a \((\omega, n)\) MRA-code without secrecy, if \(P_I = 1/\sqrt{\ell}\) then \(P_S \geq 1/\sqrt{\ell}\). Equality holds if and only if

\[
P(R_{i_1}, \ldots, R_{i_k} \text{ accept } m' | R_{i_1}, \ldots, R_{i_k} \text{ accept } m) = 1/\ell
\]

\[
P(R_j \text{ accepts } m' | R_j \text{ accepts } m) = 1/\sqrt{\ell}
\]

for \(\forall R_j, \forall m\) and \(\forall m'\) such that the source state of \(m\) is different from that of \(m'\).

**Theorem 2.3** ([38]) In a \((\omega, n)\) MRA-code without secrecy, if \(P_I = P_S = 1/\sqrt{\ell}\) then \(|E_i| \geq (\sqrt{\ell})^2\) for any \(j\). If equality holds, then each rule of \(E_j\) is used with equal probability.

KO characterized Cartesian MRA-codes that satisfy \(P_I = P_S = 1/\sqrt{\ell}\) and observed that DFY polynomial construction is an optimal construction in terms of number of keys for the transmitter and the receivers and size of the authenticator.

Definition 2.2 does not specify the relationship between the encoding functions of the transmitter and the receivers. It only requires independence of receivers’ keys for any set of \(\omega\) receivers. This independence is sufficient to ensure that the probability of success in impersonation attack by any \(\omega - 1\) receivers against another receiver is the same as that by an (outside) opponent. We give a general definition of MRA-codes in terms of commutative mappings, and for \((\omega, n)\) MRA-codes only require the success probability of attackers in impersonation and/or substitution attacks to be less than one. However we do allow coalition of insiders to have higher chance of success compared to an outsider.

### 2.3 Model and bounds

In this section we provide a formal model for MRA-codes and same lower bound. An MRA-System has three phases:

- **Key distribution:** The KDC (Key Distribution Center) privately transmits the key information to the sender and each receiver (the sender can also be the KDC).

- **Broadcast:** For a source state the sender generates the authenticated message using his/her key and broadcasts the authenticated message.
Chapter 2. Multireceiver authentication codes (MRA): Model and Bounds

- **Verification:** Each user can verify the authenticity of the broadcast message.

Denote by $X_1 \times \cdots \times X_n$ the direct product of sets $X_1, \ldots, X_n$, and by $p_i$ the projection mapping of $X_1 \times \cdots \times X_n$ on $X_i$. That is, $p_i : X_1 \times \cdots \times X_n \to X_i$ defined by $p_i(x_1, x_2, \ldots, x_n) = x_i$. Let $g_1 : X_1 \to Y_1$ and $g_2 : X_2 \to Y_2$ be two mappings, we denote the direct product of $g_1$ and $g_2$ by $g_1 \times g_2$, where $(g_1 \times g_2)(x_1, x_2) = (g_1(x_1), g_2(x_2))$. The identity mapping on a set $X$ is denoted by $1_X$.

**Definition 2.3** Let $C = (S, M, E, f)$ and $C_i = (S, M_i, E_i, f_i), i = 1, 2, \ldots, n$, be authentication codes. We call $(C; C_1, C_2, \ldots, C_n)$ a multireceiver authentication code (MRA-code) if there exist two mappings $\tau : E \to E_1 \times \cdots \times E_n$ and $\pi : M \to M_1 \times \cdots \times M_n$ such that for any $(s, e) \in S \times E$ and any $1 \leq i \leq n$ the following identity holds

$$p_i(\pi f(s, e)) = f_i((1_S \times p_i \tau)(s, e)).$$

Let $\tau_i = p_i \tau$ and $\pi_i = p_i \pi$ then we have for each $(s, e) \in S \times E$

$$\pi_i f(s, e) = f_i((1_S \times \tau_i)(s, e)).$$

We assume that for each $i$ the mapping $\tau_i : E \to E_i$ and $\pi_i : M \to M_i$ are surjective. We also assume that for each code $C_i$ the probability distribution on the source states of $C_i$ is the same with that in the A-code $C$, and the probability distribution on $E_i$ is derived from that of $E$ and the mapping $\tau_i$.

Let $T$ denote the sender and $R_1, \ldots, R_n$ denote the $n$ receivers. In order to authenticate a message, the sender and the receivers follow the following protocol:

1. The KDC (or the sender) randomly chooses a key $e \in E$ and privately transmits $e$ to $T$ and $e_i = \pi_i(e)$ to the receiver $R_i, 1 \leq i \leq n$.

2. If $T$ wants to send a source state $s \in S$ to all the receivers, $T$ computes $m = f(s, e) \in M$ and broadcasts it to all receivers.

3. Receiver $R_i$ checks whether a source state $s$ such that $f_i(s, e_i) = \pi_i(m)$ exists. If such an $s$ exists, the message $m$ is accepted as authentic. Otherwise $m$ is rejected.
Chapter 2. Multireceiver authentication codes (MRA): Model and Bounds

We adopt the Kerckhoff’s principle that everything in the system except the actual keys of the sender and receivers is public. This includes the probability distribution of the source states and the sender’s keys. From Definition 2.3 we know that the probability distribution of the sender’s key induces a probability distribution on each receiver’s key.

Attackers could be outsiders who do not have access to any key information, or insiders who have some key information. We only need to consider the latter group as it is at least as powerful as the former. We consider the systems that protect against the coalition of groups of up to a maximum size of receivers, and study impersonation and substitution attacks.

Assume there are $n$ receivers $R_1, \ldots, R_n$. Let $L = \{i_1, \ldots, i_\ell\} \subseteq \{1, \ldots, n\}$, $E_L = E_{i_1} \times \ldots \times E_{i_\ell}$ and $R_L = \{R_{i_1}, \ldots, R_{i_\ell}\}$. We consider the attack from $R_L$ on a receiver $R_i$, where $i \notin L$.

**Impersonation Attack:** $R_L$, after receiving their secret keys, send a message $m$ to $R_i$. $R_L$ is successful if $m$ is accepted by $R_i$ as authentic. We denote by $P_{I}[i, L]$ the success probability of $R_L$ in performing an impersonation attack on $R_i$. This can be expressed as

$$P_{I}[i, L] = \max_{e_L \in E_L} \max_{m \in M} P(m \text{ is accepted by } R_i | e_L) \quad (2.1)$$

where $i \notin L$.

**Substitution Attack:** $R_L$, after observing a message $m$ that is transmitted by the sender, replace $m$ with another message $m'$. $R_L$ is successful if $m'$ is accepted by $R_i$ as authentic. We denote by $P_{S}[i, L]$, the success probability of $R_L$ in performing a substitution attack on $R_i$. We have,

$$P_{S}[i, L] = \max_{e_L \in E_L} \max_{m \in M} \max_{m' \neq m \in M} P(R_i \text{ accepts } m' | m, e_L) \quad (2.2)$$

The following two bounds are generalizations of Simmons’ bound [58] and Brickell’s bound [7], when the attack is from a group of insiders who have access to part of the key information.

**Theorem 2.4** Let $P_{I}[i, L]$ and $P_{S}[i, L]$ be defined as in equation (2.1) and (2.2). Assume that $M \neq M'$, then

1. $P_{I}[i, L] \geq 2^{-I(M:E_i|E_L)}$.

2. $P_{S}[i, L] \geq 2^{-I(M':E_i|M,E_L)}$. 


Chapter 2. Multireceiver authentication codes (MRA): Model and Bounds

The proof is given in the Appendix A.

**Corollary 2.1**

\[ P_S[i, L] \geq 2^{-H(E_i|M, E_L)}. \]

**Proof:** The corollary follows from Theorem 2.4 by noting that \( I(M'; E_i|M, E_L) = H(E_i|M, E_L) - H(E_i|M', M, E_L) \).

A \((\omega, n)\) **MRA-code** is an MRA-code in which there are \(n\) receivers such that no subset of \(\omega - 1\) receivers can construct a fraudulent codeword accepted by another receiver. We note that in this definition, the only requirement is that the chance of success of the attackers is less than one but it is possible that some coalition of attackers have a better chance of success than an outsider.

A \((\omega, n)\) **MRA-code is perfect for impersonation** if the chance of success of any group of up to \(\omega - 1\) receivers in an impersonation attack is the same as an outsider. Similarly, a \((\omega, n)\) **MRA-code is perfect for substitution** if the chance of success for any group of up to \(\omega - 1\) receivers in a substitution attack is the same as an outsider.

**Lemma 2.1** A sufficient condition for a \((\omega, n)\) **MRA-code** to be perfect for impersonation is that \( P(e_i|e_L) = P(e_i) \) for all \(\omega\)-subsets \(L \cup \{i\}, i \neq L\) of \(\{1, \ldots, n\}\).

**Proof:** Consider the \(A\)-code \(C_i = (S, M_i, E_i)\), we define an authentication function \(X(m_i, e_i)\) on \(M_i \times S_i\) as

\[
X_S(m_i, e_i) = \begin{cases} 
1 & \text{if } m_i \text{ is authentic for the key } e_i \\
0 & \text{otherwise.} 
\end{cases}
\]

We have \(P(\pi_i(m))\) is valid in \(C_i = \sum_{e_i \in E_i} X(\pi_i(m), e_i)P(e_i)\). By definition of \(X_I(m, e_i, e_L)\) (see Appendix A), we know that for any given \(e_L\) in accordance with \(\tau_L = e_L\) and \(\tau_i = e_i\), \(X(\pi_i(m), e_i) = X_I(m, e_i, e_L)\). Thus we have

\[
P_I[i, L] = \max_{m \in M} P(m \text{ is accepted by } R_i|e_L) = \max_{m \in M} \sum_{e_i \in E_i} X_I(m, e_i, e_L)P(e_i|e_L) = \max_{m \in M} \sum_{e_i \in E_i} X(\pi_i(m), e_i)P(e_i|e_L) = P_I[i]
\]
In the above Lemma, \( P_I[i] \) is the success probability of an outsider in impersonation attack and is given by

\[
P_I[i] = \max_{m \in M} P(R_i \text{ accepts } m) = \max_{m \in M} P(\pi_i(m) \text{ is valid in } C_i).
\]

It should also be noted that a \((\omega, n)\) MRA-code which is perfect for impersonation is not necessarily perfect for substitution.

Let \((C; C_1, \ldots, C_n)\) be an MRA-code. Define \( P_I \) and \( P_S \) as follow

\[
P_I = \max_{L \cup \{i\}} \{P_I[i, L]\}
\]

\[
P_S = \max_{L \cup \{i\}} \{P_S[i, L]\}
\]

where the maximum is taken over all possible \( \omega \)-subsets \( L \cup \{i\} \) (\( i \not\in L \)) of \( \{1, 2, \ldots, n\} \).

In other words, \( P_I \) and \( P_S \) are the best chances for a group of \( \omega - 1 \) receivers to succeed in impersonation or substitution attacks against a single receiver, respectively.

We define the deception probability of a \((\omega, n)\) MRA-system as \( P_D = \max\{P_I, P_S\} \).

**Theorem 2.5** Let \((C; C_1, \ldots, C_n)\) be a \((\omega, n)\) MRA-code. Assume that \( P_D \leq 1/q \) and there is a uniform probability distribution on the source states \( S \). Then

1. \( |E_i| \geq q^2 \), for each \( i \in \{1, \ldots, n\} \).
2. \( |E| \geq q^{2\omega} \).
3. \( |M| \geq q^{\omega} |S| \).

The bound are tight and there exists a system that satisfies the bounds with equality.

**Proof:**

1. For each \((\omega - 1)\)-subset \( L \) of \( \{1, \ldots, n\} \) where \( i \not\in L \), by Theorem 2.4 and Corollary 2.1 we have

\[
\left( \frac{1}{q} \right)^2 \geq P_D \geq P_I[i, L]P_S[i, L] \geq 2^{-(I(M; E_i|E_L) + H(E_i|E_L, M))} = 2^{-H(E_i|E_L)} \geq 2^{-H(E_i)} \geq 2^{-\log |E_i|} = \frac{1}{|E_i|}.
\]

It follows that \( |E_i| \geq q^2 \).
Chapter 2. Multireceiver authentication codes (MRA): Model and Bounds

2. Assume that \( L_i = \{1, \ldots, i - 1, i + 1, \ldots, \omega\}, \) \( i = 1, \ldots, \omega. \) We have,

\[
\left(\frac{1}{q}\right)^{2\omega} \geq \prod_{i=1}^{\omega} P_{L_i}[i, L_i]P_{S}[i, L_i] \geq 2^{\sum_{i=1}^{\omega} H(E_i|E_{L_i})} \\
\geq 2^{-\sum_{i=1}^{\omega} H(E_1, \ldots, E_{i-1})} = 2^{-H(E_1, \ldots, E_\omega)} \\
\geq 2^{-H(E)} \geq 2^{-\log|E|} = \frac{1}{|E|},
\]

therefore, \( |E| \geq q^{2\omega}. \)

3. Since \( \tau : E \to E_1 \times \cdots \times E_\omega \) indices a mapping from \( E \) to \( E_1 \times \cdots \times E_\omega, \) we have \( I(M; E) \geq I(M; E_1, \ldots, E_\omega). \) It follows that

\[
2^{-I(M; E)} \leq 2^{-I(M; E_1, \ldots, E_\omega)} \\
= 2^{-\sum_{i=1}^{\omega} I(M; E_i|E_1, \ldots, E_\omega)} \\
= 2^{-\sum_{i=1}^{\omega} I(M; E_i|E_1, \ldots, E_{i-1})} \\
= \prod_{i=1}^{\omega} 2^{-I(M; E_i|E_1, \ldots, E_{i-1})} \leq \prod_{i=1}^{\omega} P[i, Q_i],
\]

where \( Q_i = \{1, \ldots, i-1\}. \) Since for each \( 1 \leq i \leq \omega, \) we have \( P[i, Q_i] \leq P[i, L_i] \leq \frac{1}{q}, \) it follows that,

\[
2^{-I(M; E)} = 2^{-(H(M)-H(M|E))} = 2^{-H(M)}2^{H(M|E)} \leq \left(\frac{1}{q}\right)^\omega.
\]

Since \( S \) is assumed to be uniformly distributed, we know that \( H(M|E) = H(S) = \log |S|. \) Hence, \( |M| = 2^{\log |M|} \geq 2^{H(M)} \geq q^{\omega}|S| \) which proves (iii).

\[\blacksquare\]
Chapter 3

Some Constructions of MRA-code

3.1 The DFY polynomial construction

In [25], Desmedt, Frankel and Yung gave two constructions for MRA-codes: one is based on polynomials and the other is based on finite geometries. We briefly review DFY’s polynomial construction because generalizations of this scheme will be discussed in the following Sections of this thesis. Detail on the geometric construction can be found in [25].

Assume there is a sender $T$ and $n$ receivers $R_1, R_2, \ldots, R_n$. DFY polynomial scheme works as follows. The key for $T$ consists of two random polynomials $P_0(x)$ and $P_1(x)$, each of degree at most $\omega - 1$, with coefficients in $GF(q)$, where $q > \max\{|S|, n\}$. The key for $R_i$ consists of $P_0(i)$ and $P_1(i)$. For a source state $s \in GF(q)$, $T$ broadcasts $(s, A(x))$ where $A(x) = P_0(x) + sP_1(x)$. $R_i$ accepts $(s, A(x))$ as authentic if $A(i) = P_0(i) + sP_1(i)$. It is proved [25] that the construction results in a MRA-code with $P_D = 1/q$ and the following parameters:

$$|S| = \frac{1}{q}$$

$$|E_i| = q^2 \quad \forall i \in \{1, \ldots, n\}$$

$$|E| = 2^{2\omega}$$

$$|M| = q^{\omega}|S|.$$  

Hence the bound in Theorem 2.5 can be achieved with equality.

A trivial construction for MRA-codes requires the sender to store many key bits and produces a long tag for the authenticated message. DFY scheme significantly reduces the
size of the key storage and the length of the authentication tag. However, the order of the
field $GF(q)$ must be chosen bigger than the size of the source space and the number of
the receivers. Indeed $q$, which can be thought of as the security parameter of the system,
$P_I = P_S = 1/q$, determines the size of the key storage and the length of the authentication
tag. This make the construction very restrictive because although it is acceptable to have
the key storage, and the length of the tag, a function of the security parameter of the
system, but having the number of the receivers and the size of the source bounded by it,
is not reasonable. In particular, when the size of the source or the number of the receivers
are very large, $P_I$ and $P_S$ will be unnecessarily small and the key storage of the sender and
the receivers, along with the length of the authentication tag will become prohibitively
large.

In practice, we may deal with the scenarios where we are satisfied with deception
probabilities higher than $1/q$, but key storage or communication bandwidth are bounded.

3.2 Generalizations

The basic MRA-code, can be generalized in a number of ways. In this Section we look at
two possible generalizations.

3.2.1 MRA-code for multiple message transmissions

In the basic model of MRA-codes, security analysis is for a single message transmission
(only impersonation and substitution attacks are considered) and for a second message
no protection is guaranteed. To provide protection for multiple message transmission one
possibility is to use a new key after each message is broadcasted. This is a very inefficient
solution both in terms of going through a key distribution phase after each message and
in term of the amount of key information required for each message. We present a system
that uses a single key distribution phase for multiple message transmission, and requires
less key information per communicated message.

Generalized DFY scheme for multiple messages

Assume messages are distinct and $h < |S|$. The scheme consists of the following steps:
1. **Key distribution**: The KDC randomly generates $h+1$ polynomials $P_0(x), P_1(x), \ldots, P_h(x)$ of degree at most $\omega - 1$ and chooses $n$ distinct elements $x_1, x_2, \ldots, x_n$ of $GF(q)$. KDC makes $x_i$’s public and sends privately $(P_0(x), P_1(x), \ldots, P_h(x))$ to the sender $T$, and $(P_0(x_i), P_1(x_i), \ldots, P_h(x_i))$ to the receiver $R_i$.

2. **Broadcast**: For a source state $s$, $T$ computes $A_s(x) = P_0(x) + sP_1(x) + \cdots + s^hP_h(x)$ and broadcast $(s, A_s(x))$.

3. **Verification**: $R_i$ accepts $(s, A_s(x))$ as authentic if $A_s(x_i) = P_0(x_i) + sP_1(x_i) + \cdots + s^hP_h(x_i)$.

The above scheme is a multi-receiver authentication code in which each key can be used to authenticate up to $h$ messages. To prove the security of the scheme, we consider the scenario where for a given key $(P_0(x), P_1(x), \ldots, P_h(x))$, $h$ source states $s_1, s_2, \ldots, s_h$ have been authenticated and there are $\omega - 1$ receivers who want to construct a fraudulent codeword that is acceptable by one of the other receivers. Without loss of generality, we may assume that the malicious receivers are $R_1, R_2, \ldots, R_{\omega-1}$.

Let $P_i(x) = a_{i0} + a_{i1}x + \cdots + a_{i\omega-1}x^{\omega-1}$, $0 \leq i \leq h$. Since $s_1, s_2, \ldots, s_h$ have been sent, $A_{s_1}(x), A_{s_2}(x), \ldots, A_{s_h}(x)$ are publicly known where

$$A_{sj}(x) = b_{j0} + b_{j1}x + \cdots + b_{j\omega-1}x^{\omega-1}, \text{ for all } 1 \leq j \leq h.$$ 

and the $\omega - 1$ receivers $R_1, R_2, \ldots, R_{\omega-1}$ know their keys

$$(P_0(x_1), P_1(x_1), \ldots, P_h(x_1)), \ldots, (P_0(x_{\omega-1}), P_1(x_{\omega-1}), \ldots, P_h(x_{\omega-1})).$$

It follows that the malicious receivers know the following two matrix equations

$$\begin{pmatrix}
    a_{00} & a_{10} & \cdots & a_{h0} \\
    a_{01} & a_{11} & \cdots & a_{h1} \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{0\omega-1} & a_{1\omega-1} & \cdots & a_{h\omega-1}
\end{pmatrix}
\begin{pmatrix}
    1 & 1 & \cdots & 1 \\
    s_1 & s_2 & \cdots & s_h \\
    \vdots & \vdots & \ddots & \vdots \\
    s_h^1 & s_h^2 & \cdots & s_h^h
\end{pmatrix}
= \begin{pmatrix}
    b_{10} & b_{11} & \cdots & b_{1\omega-1} \\
    b_{20} & b_{21} & \cdots & b_{2\omega-1} \\
    \vdots & \vdots & \ddots & \vdots \\
    b_{h0} & b_{h1} & \cdots & b_{h\omega-1}
\end{pmatrix}$$

and
Chapter 3. Some Constructions of MRA-code

\[
\begin{pmatrix}
1 & x_1 & \cdots & x_1^{\omega-1} \\
1 & x_2 & \cdots & x_2^{\omega-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & x_{\omega-1} & \cdots & x_{\omega-1}^{\omega-1}
\end{pmatrix}
\begin{pmatrix}
a_{00} & a_{10} & \cdots & a_{h0} \\
a_{01} & a_{11} & \cdots & a_{h1} \\
\vdots & \vdots & \ddots & \vdots \\
a_{0\omega-1} & a_{1\omega-1} & \cdots & a_{h\omega-1}
\end{pmatrix}
= 
\begin{pmatrix}
P_0(x_1) & \cdots & P_h(x_1) \\
P_0(x_2) & \cdots & P_h(x_2) \\
\vdots & & \ddots & \vdots \\
P_0(x_{\omega-1}) & \cdots & P_h(x_{\omega-1})
\end{pmatrix}
\]

The above two equations can be rewritten as

\[
AM_h = B \quad (3.1)
\]
\[
X_{\omega-1}A = C \quad (3.2)
\]

where \(A, M_h, B, X_{\omega-1}\) and \(C\) denote the corresponding matrices in an obvious manner.

We first give a lemma, which says that knowing \(M_h, B, X_{\omega-1}\) and \(C\) cannot determine \(A\).

In other words, the matrix satisfying (3.1) and (3.2) is not unique.

**Lemma 3.1** There exists \(q\) different matrices \(D\) such that \(DM_h = B\) and \(X_{\omega-1}D = C\)

**Proof:** It is sufficient to prove that there exist \(q\) different matrices \(D\) such that \(DM_h = 0\) and \(X_{\omega-1}D = 0\). First we observe that given an \(n \times m\) matrix \(D_0 = (d_{ij})\), we can associate it a polynomial over \(x\) and \(y\)

\[
P(x, y) = \left(1, x, x^2, \cdots, x^{n-1}\right) D_0
\]

and conversely, every polynomial \(P(x, y)\) can be written in the form (3.3) for some \(n \times m\) matrix \(D_0\). Now consider the polynomial

\[
P(x, y) = (x - x_1)(x - x_2) \cdots (x - x_{\omega-1})(y - s_1)(y - s_2) \cdots (y - s_h).
\]

Let \(P(x, y) = \left(1, x, x^2, \cdots, x^{\omega-1}\right) D\), where \(D\) is a \(k \times (h + 1)\) matrix and \(D \neq 0\).

Clearly, \(P(x_1, y) = P(x_2, y) = \cdots = P(x_{\omega-1}, y) = 0\) for all \(y\). It follows
Indeed, we may choose \((h+1)\) distinct elements \(y_1, y_2, \ldots, y_{h+1}\) in \(GF(q)\), then

\[
P(x_1, y_i) = P(x_2, y_i) = \cdots = P(x_{\omega-1}, y_i) = 0
\]

for all \(1 \leq i \leq h+1\). Thus we have

\[
\begin{pmatrix}
1 & x_1 & \cdots & x_1^{\omega-1} \\
1 & x_2 & \cdots & x_2^{\omega-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & x_{\omega-1} & \cdots & x_{\omega-1}^{\omega-1}
\end{pmatrix}
D
\begin{pmatrix}
1 & 1 & \cdots & 1 \\
y_1 & y_2 & \cdots & y_{h+1} \\
\vdots & \vdots & \ddots & \vdots \\
y_1^h & y_2^h & \cdots & y_{h+1}^h
\end{pmatrix}
= 0.
\]

Since

\[
\begin{pmatrix}
1 & 1 & \cdots & 1 \\
y_1 & y_2 & \cdots & y_{h+1} \\
\vdots & \vdots & \ddots & \vdots \\
y_1^h & y_2^h & \cdots & y_{h+1}^h
\end{pmatrix}

\]

is a Vandermonde matrix, the desired result follows. Similarly, we have

\[
D
\begin{pmatrix}
1 & 1 & \cdots & 1 \\
s_1 & s_2 & \cdots & s_{h+1} \\
\vdots & \vdots & \ddots & \vdots \\
s_1^h & s_2^h & \cdots & s_{h+1}^h
\end{pmatrix}
= 0.
\]

For each \(r \in GF(q)\), we also have \((rD)M_h = 0\) and \(X_{\omega-1}(rD) = 0\). Thus there are \(q\) different matrices \(\{rD\mid r \in GF(q)\}\) with the desired property. So we complete the proof of the lemma.

\section*{Theorem 3.1} The above scheme is a \((\omega, n)\) unconditionally secure multi-receiver authentication code in which every key can be used to authenticate up to \(h\) messages.

\section*{Proof} We only consider the substitution attack, the proof for the impersonation is similar. The malicious receivers \(P_1, \ldots, P_{\omega-1}\) want to generate a valid codeword \((s_{h+1}, b_0 + \ldots, b_n)\).
$b_1x + \cdots + b_{\omega-1}x^{\omega-1}$) such that it is accepted by $R_\omega$. What they try to do is to guess the value of $P_0(x_\omega) + s_{h+1}P_1(x_\omega) + \cdots + s_{h+1}^hP_h(x_\omega)$ and construct a polynomial $A(x) = b_0 + b_1x + \cdots + b_{\omega-1}x^{\omega-1}$ such that

$$b_0 + b_1x + \cdots + b_{\omega-1}x^{\omega-1} = P_0(x_\omega) + s_{h+1}P_1(x_\omega) + \cdots + s_{h+1}^hP_h(x_\omega).$$

In the following we will show that the information held by the colluders allows them to calculate $q$ equally likely different tags for $s_{h+1}$ and, hence, their probability of success is $1/q$.

From Lemma 3.1 we know that there are $q$ different matrices $D$ such that $DM_h = B$ and $X_{\omega-1}D = C$. This implies that there are $q$ different $(h+1)$-tuples of polynomials $(Q_0(x), Q_1(x), \ldots, Q_{h+1}(x))$ such that each of them is equally likely to be the key of the sender.

Now we note that: (1) the $q$ different $(h+1)$-tuples of polynomials give rise to $q$ different possible keys for $R_\omega$. Indeed, suppose that $Q[x] = (Q_0(x), Q_1(x), \ldots, Q_{h+1}(x))$ and $Q'[x] = (Q'_0(x), Q'_1(x), \ldots, Q'_{h+1}(x))$ are two different keys of $T$, and their corresponding matrix are $D$ and $D'$. Then

$$C_k = (Q_0(x_\omega), Q_1(x_\omega), \ldots, Q_{h+1}(x_\omega)) = (1, x_\omega, \ldots, x_\omega^{\omega-1})D$$

and

$$C'_k = (Q'_0(x_\omega), Q'_1(x_\omega), \ldots, Q'_{h+1}(x_\omega)) = (1, x_\omega, \ldots, x_\omega^{\omega-1})D',$$

are their corresponding keys for $R_\omega$. By Lemma 3.1 we know that $X_{\omega-1}D = X_{\omega-1}D' = C$. It follows that $X_\omega D = \begin{pmatrix} C \\ C_\omega \end{pmatrix}$ and $X_\omega D' = \begin{pmatrix} C \\ C'_\omega \end{pmatrix}$. The matrix $X_\omega$ is a Vandermonde matrix and so it is invertible. Using the assumption that $D \neq D'$, we get that $C_k \neq C'_k$.

Next, (2) we prove that $C_k(1, s_{h+1}, \ldots, s_{h+1}^h)^T \neq C'_k(1, s_{h+1}, \ldots, s_{h+1}^h)^T$, where $G^T$ denotes the transpose of the matrix $G$. Again, by Lemma 3.1, we have $C_\omega M_h = C'_\omega M_h$. Now if $C_k(1, s_{h+1}, \ldots, s_{h+1}^h)^T \neq C'_k(1, s_{h+1}, \ldots, s_{h+1}^h)^T$ then $C_\omega M_{h+1} = C'_\omega M_{h+1}$. But since $M_{h+1}$ is also invertible, it follows that $C_\omega = C'_\omega$, which is a contradiction.

Combining (1) and (2), we have that, for a given $s_{h+1}$, there are $q$ different values $C_k(1, s_{h+1}, \ldots, s_{h+1}^h)^T$, all equally likely to be acceptable by $R_\omega$. Hence, the probability that the $\omega - 1$ receivers correctly guess $A(x)$ is $1/q$. 

\[ \square \]
3.2.2 MRA-codes with dynamic sender

An interesting extension of the model of MRA-code is when the sender is not fixed and can be any member of the group. In this case, key distribution is performed by a Trusted Authority (TA) who is only active during key distribution phase. We call the system MRA-code with dynamic sender. There are many applications for such systems. For example, providing authentication in group communication where members of a group want to broadcast messages such that every other group member can verify the authenticity of the received messages. It is worth noting that providing authentication in group communication is much more difficult than providing confidentiality because in the former group members can participate in coordinated attack against other group members while in the latter protection is only provided against outsider’s eavesdropping.

Allowing the sender to be dynamic introduces the notion of authenticating with respect to a particular identity. That is, to verify authenticity of a received message a receiver must first assume an identity for the sender and then verify the message with respect to this particular sender. An authenticated message in general carries information that indicates its origin, along with its content information. Hence, the system must provide both origin (entity) authentication and message authentication. In other words, the success of an attacker(s) could be by replacing the identity information, or the message content.

The model

In the model of MRA-code with dynamic sender, there are $n$ users $P = \{P_1, \ldots, P_n\}$, who want to communicate over a broadcast channel. The channel is subject to spoofing attack: that is a codeword can be inserted into the channel or, a transmitted codeword can be substituted with a fraudulent one. An attack is directed towards a channel, consisting of a pair of users $\{P_i, P_j\}$, where $P_i$ is the sender and $P_j$ is the receiver. A spoofer might be an outsider, or a coalition of $\omega - 1$ users. The aim of the spoofer(s) is to construct a codeword that $P_j$ accepts as being sent from $P_i$. We assume that the TA is only active during key distribution phase. The system has three phases:

1. Key Distribution: The TA generates and distributes secret information to each user.
2. **Broadcast**: One of the users generates an authenticated message for a source state of his/her choice, and broadcasts it.

3. **Verification**: Every user can verify authenticity of the broadcasted message using their own secret information.

**Definition 3.1** A \( (\omega, n) \) MRA-code with dynamic sender is a code for which no \( \omega - 1 \) subsets of users can perform impersonation and/or substitution attack on any other pair of users.

For the sake of simplicity, we assume that after the key distribution phase, each user can only send at most a single authenticated message.

From the above definition, we make the following observations:

1. In a \( (\omega, n) \) MRA-code with dynamic sender during the key distribution phase, the TA does not know which user is going to broadcast. That is, there are \( n \) users and everyone of them could be a sender.

2. A \( (\omega, n) \) MRA-code with dynamic sender is a \( (\omega', n) \) MRA-code with dynamic sender for any \( \omega' \leq \omega \).

3. We assume that a message is sent only once by a single sender. Therefore, a possible attack is to change the origin information of codeword and leave the message content intact.

A straightforward construction based on conventional A-codes is to give each pair of users, \((P_i, P_j)\), a shared secret key. Note that now a user can generate the authenticators for a message using the secret keys he shares with all \( P_j \)'s, and broadcast the concatenation of them. In this case there are \( n(n-1)/2 \) pairs of users, which means that a user has to store \( (n-1) \) keys, and the TA has to generate and store \( (n-1)n/2 \) keys. The disadvantages of this scheme are the large amount of keys stored by each user, together with the long tag for the authenticated message. Our aim is to give more efficient constructions which reduce the key management of both the TA and the users, and reduce the authenticator size, compared to this trivial scheme.
Lower Bounds

To define $P_I$ and $P_S$ in an MRA-code with dynamic sender, we note that because every user can be a sender, when a message is received by a user $P_i$, he/she must first assume an identity for the sender and then verify the authenticity of the message with respect to the assumed identity. The enemy is a set of $\omega - 1$ malicious users, $P_{l_1}, \ldots, P_{l_{\omega-1}}$, who attack a pair of other users. For example, targeting the pair $\{P_i, P_j\}$, results in $P_j$ accepting a fraudulent message as being sent from $P_i$. In an impersonation attack, $P_{l_1}, \ldots, P_{l_{\omega-1}}$ collude and try to launch an attack against a pair of users $P_i$ and $P_j$, by generating a message such that $P_j$ accepts it as authentic and being sent from $P_i$. We denote the success probability in this case by $P_I[m; i, j; L]$, where $L = \{P_{l_1}, \ldots, P_{l_{\omega-1}}\}$. $P_I$ is the best probability of success in such attacks and is defined by

$$P_I = \max_{L, i, j} \max_m P_I[m; i, j; L]$$

where $L \cup \{i, j\}$ runs through all $(\omega + 1)$-subsets of $\{1, 2, \ldots, n\}$.

In substitution attack, there are two distinct cases:

1. **Message substitution:** After seeing a valid message $m$ broadcasted by $P_i$, the users $\{P_{l_1}, \ldots, P_{l_{\omega-1}}\}$ construct a new message $m'$ ($m \neq m'$) such that $P_j$ will accept $m'$ as being sent from $P_i$. We denote the success probability in this case by $P_{S_{\text{message}}}[m, m'; i, j; L]$, and the best probability of such an attack is denoted by $P_{S_{\text{message}}}$:

$$P_{S_{\text{message}}} = \max_{L, i, j} \max_{m' \neq m} P_{S_{\text{message}}}[m, m'; i, j; L]$$

where $L \cup \{i, j\}$ runs through all $\omega + 1$-subsets of $\{1, 2, \ldots, n\}$.

2. **Entity substitution:** After seeing a valid message $m$ broadcasted by $P_i$, the users $\{P_{l_1}, \ldots, P_{l_{\omega-1}}\}$ construct a new message $m'$, not necessarily different from $m$, such that $P_j$ will accept $m'$ as being sent from $P_{i'}$, where $i \neq i'$. We denote the success probability in this case by $P_{S_{\text{entity}}}[m, m'; i, i', j; L]$, and the best probability of such an attack by

$$P_{S_{\text{entity}}} = \max_{L, i, i', j} \max_{m' \neq m} P_{S_{\text{entity}}}[m, m'; i, i', j; L],$$

where $L \cup \{i, i', j\}$ runs through all $(\omega + 2)$-subsets of $\{1, 2, \ldots, n\}$.
Now the probability of success in the substitution attack for the whole system is defined as

\[ P_S = \max\{P_{message}, P_{entity}\}. \]

**Theorem 3.2** In a \((\omega, n)\) MRA-code with dynamic sender, assume that \( P_I = P_S \leq 1/q \) and assume there is a uniform probability distribution on the source states \( S \). Then we have:

1. \(|E_i| \geq q^{2\omega}\), for each \( i \in \{1, 2, \ldots, n\}\),
2. \(|M_i| \geq q^{\omega}|S|\), for each \( i \in \{1, 2, \ldots, n\}\),

where \( E_i \) is the set of possible keys of \( P_i \) and \( M_i \) is the set of possible codewords when \( P_i \) is a sender, for all \( i \in \{1, 2, \ldots, n\}\). These are tight bounds and there exists a system that satisfies them with equality.

**Proof:** For each \( i, 1 \leq i \leq n \), \( P_i \) is a possible sender and so the \((\omega, n)\) MRA-system with dynamic sender induces a \((\omega, n - 1)\) MRA-code, in which the probability of success in impersonation and substitution attacks are both \( 1/q \). By applying Theorem 2.5, we obtain the required results.

\[ \square \]

In the next Section we will show that the bounds are tight by giving a construction that meets them.

**An optimal construction**

Now we give a construction for a \((\omega, n)\) MRA-code with dynamic sender, which is based on symmetric polynomials in two variables. In [53] a \((\omega, n)\) MRA-code with dynamic sender using Blom’s key distribution scheme is proposed. The following construction is a slightly modified version of the construction given in [53]. We show that the construction has the minimum length of keys for users and the authenticator, and it meets the bounds in Theorem 3.2 with equality. We first briefly review Blom’s key distribution scheme.
Blom’s Key distribution scheme

Let \( q \geq n \) be a prime power. The TA randomly chooses a symmetric polynomial, \( F(x, y) \), with coefficients in \( GF(q) \) and of degree less than \( \omega \). For \( 1 \leq i \leq n \), the TA computes the polynomial \( G_i(x) = F(x, i) \) and gives \( G_i(x) \) to user \( P_i \), i.e. \( G_i(x) \) is the secret information of \( P_i \). The key associated with the pair of users \( P_i \) and \( P_j \) is calculated as, \( k_{ij} = G_i(j) = G_j(i) \). It is proved [6] that the scheme is unconditionally secure against the collusion of \( \omega - 1 \) users in the following sense: the coalition of any \( \omega - 1 \) out of \( n \) users, say \( P_{i_1}, \ldots, P_{i_{\omega - 1}} \), has no information about the key \( k_{ij} \) for the pair \( i, j \) where \( i, j \notin \{i_1, \ldots, i_\omega\} \).

\((\omega, n)\) MRA-code with dynamic sender based on Blom’s scheme

The \((\omega, n)\) MRA-code with dynamic sender based on the Blom’s scheme, works as follows. Let \( S \) be the set of source states and \( q \geq \max\{|S|, n\} \) be a prime power.

1. **Key distribution:** The TA chooses \( n \) distinct numbers \( a_i \) in \( GF(q) \) (associates \( a_i \) to user \( P_i \), \( 1 \leq i \leq n \)). These values are public and are used as identity information for users. Then, the TA randomly chooses two symmetric polynomials of degree less than \( \omega \) with coefficients in \( GF(q) \),

\[
P_\ell(x, y) = \begin{pmatrix} 1 \\ y \\ y^2 \\ \vdots \\ y^{\omega - 1} \end{pmatrix} A_\ell
\]

where \( \ell = 0, 1 \) and \( A_\ell \) is a \( \omega \times \omega \) symmetric matrix for \( \ell = 0, 1 \). For \( 1 \leq i \leq n \), the TA computes the polynomials

\[
G_{\ell i} = P_\ell(x, a_i) = \begin{pmatrix} 1 \\ a_i \\ a_i^2 \\ \vdots \\ a_i^{\omega - 1} \end{pmatrix} A_\ell
\]

for \( \ell = 0, 1 \) and gives the 2 tuple of polynomials, \((G_{0i}(x), G_{1i}(x))\), to user \( P_i \). This constitutes the secret information of \( P_i \).
2. **Broadcast:** For \(1 \leq i \leq n\), assume that user \(P_i\) wants to generate the authenticated message for a source state \(s \in S\). \(P_i\) computes the polynomial \(M_i(x) = G_{0i}(x) + sG_{1i}(x)\) and broadcast \((s, a_i, M_i(x))\).

3. **Verification:** The user \(P_j\) can verify the authenticity of the message in the following way. \(P_j\) accepts \((s, a_i, M_i(x))\) as authentic and being sent from \(P_i\) if \(M_i(a_j) = G_{0j}(a_i) + sG_{1j}(a_i)\).

**Theorem 3.3** The above scheme is a \((\omega, n)\) MRA-code with dynamic sender with \(P_I = P_S = 1/q\).

**Proof:** Assume that after seeing an authenticated message \((s, a_i, M_i(x))\) broadcasted by user \(P_i\), the users \(P_1, \ldots, P_{\omega-1}\) want to generate a new message \((s', a_i, M'_i(x))\), where \(s' \neq s\) such that user \(P_j\) will accept it as authentic, i.e. \(M'_i(a_j) = G_{0j}(a_i) + s'G_{1i}(a_i)\). First, we observe that for each \(m \in GF(q)\) each user, say \(P_h\), can calculate the polynomial

\[
G_{0h}(x) + mG_{1h} = \left(1, x, x^2, \ldots, x^{\omega-1}\right) \left(A_0 + mA_1\right) \begin{pmatrix} 1 \\ a_h \\ a_h^2 \\ \vdots \\ a_h^{\omega-1} \end{pmatrix}.
\]

It follows that, for each \(m \in GF(q)\), \(P_1, \ldots, P_{\omega-1}\) can calculate a \(\omega \times (\omega - 1)\) matrix \(D[m]\) such that the following identity holds

\[
(A_0 + mA_1) \begin{pmatrix} 1 & \cdots & 1 \\ a_1 & \cdots & a_{\omega-1} \\ \vdots & \ddots & \vdots \\ a_1^{\omega-1} & \cdots & a_{\omega-1}^{\omega-1} \end{pmatrix} = D[m], \tag{3.4}
\]

Since \((s, a_i, M_i(x))\) is broadcasted it follows that \(P_1, \ldots, P_{\omega-1}\) know the following polynomial
Chapter 3. Some Constructions of MRA-code

\[ g(x) = (1, x, x^2, \ldots, x^{\omega-1}) (A_0 + mA_1) \begin{pmatrix} 1 \\ a_i \\ a_i^2 \\ \vdots \\ a_i^{\omega-1} \end{pmatrix}. \]

By combining equation 3.4 and polynomial \( g(x) \), \( P_1, \ldots, P_{\omega-1} \) can also calculate matrices \( B \) and \( C \) such that the following equations hold

\[ A_0 + sA_1 = C \quad \text{(3.5)} \]

\[ (A_0 + mA_1) \begin{pmatrix} 1 & \cdots & 1 \\ a_1 & \cdots & a_{\omega-1} \\ \vdots & \ddots & \vdots \\ a_1^{\omega-1} & \cdots & a_{\omega-1}^{\omega-1} \end{pmatrix} = D[m] \text{ for all } m \in GF(q). \quad \text{(3.6)} \]

We claim that in equations (3.5) and (3.6), knowing \( C \) and \( D[m] \) for all \( m \in GF(q) \) does not help in determining the 2-tuple matrices \( (A_0, A_1) \). Indeed, there exist \( q \) distinct 2-tuple matrices \( (A_0, A_1) \) satisfying equation (3.5) and (3.6). This is equivalent to the following statement: \textit{There exists a 2-tuple matrices } \( (A_0, A_1) \neq (0,0) \text{ such that the following equations hold} \]

\[ A_0 + sA_1 = 0 \quad \text{(3.7)} \]

\[ (A_0 + mA_1) \begin{pmatrix} 1 & \cdots & 1 \\ a_1 & \cdots & a_{\omega-1} \\ \vdots & \ddots & \vdots \\ a_1^{\omega-1} & \cdots & a_{\omega-1}^{\omega-1} \end{pmatrix} = 0 \text{ for all } m \in GF(q). \quad \text{(3.8)} \]

Indeed, consider the symmetric polynomial

\[ P(x, y) = (x - a_1) \cdots (x - a_{\omega-1})(y - a_1) \cdots (y - a_{\omega-1}) \]

\[ = (1, x, x^2, \ldots, x^{\omega-1}) A \begin{pmatrix} 1 \\ y \\ y^2 \\ \vdots \\ y^{\omega-1} \end{pmatrix}, \]
where $A$ is a $\omega \times \omega$ symmetric matrix and $A \neq 0$. We define $A_0 = -sA$ and $A_1 = A$, then it is not difficult to verify that $(A_0, A_1)$ satisfies the desired properties.

We note that since $(-sA, A)$ satisfy equations (3.7) and (3.8), then also $(-rsA, rA)$ for all $r \in GF(q)$ satisfy equations (3.7) and (3.8). This implies that there are $q$ distinct 2-tuple symmetric polynomials which are equally likely to be chosen by the TA. For each 2-tuple matrices $(A_0, A_1)$ of the form $(-rsA, rA)$, let

$$
\begin{pmatrix}
1 \\
a_i \\
a_i^2 \\
\vdots \\
\alpha_i^{\omega - 1}
\end{pmatrix}
\begin{pmatrix}
1 \\
a_i \\
a_i^2 \\
\vdots \\
\alpha_i^{\omega - 1}
\end{pmatrix} = d.
$$

Then it is straightforward to verify that $d = 0$ if and only if $r = 0$. This is equivalent to say that the $q$ distinct possible 2-tuple polynomials $(P_0(x, y), P_1(x, y))$ chosen by the TA result in $q$ distinct values of the form $P_0(a_i, a_j) + s'P_1(a_i, a_j)$. Therefore, the probability of message substitution attack $P_{\text{msg}}$ is $1/q$. Similarly, we can prove $P_{\text{ent}} = P_I = 1/q$.

In the above construction the size of each user’s key is $|E_i| = q^{2\omega}$, for all $1 \leq i \leq n$, and the size of codewords is $M_i = q^{\omega + 1} = q^{\omega} |S|$. Hence, we have shown that the bounds given in Theorem 3.2 are satisfied with equality.
Chapter 4

Unconditionally secure signatures

4.1 Towards Unconditionally secure signatures: Transferability

In the schemes presented before all messages are transmitted over a broadcast channel, and in this particular situation, transferability is not required. However, for a digital signature (for point-to-point communication), transferability is a property that cannot be neglected. That is, a signature system must allow users to pass signatures among users without compromising their integrity. Roughly speaking, dynamic MRA (DMRA) (and in general a MRA-codes) do not fulfill this requirements. As an example, we show the vulnerability of the DMRA presented in Section 3.2.2, where it allows users to pass authenticated messages among users without a broadcast channel.

Suppose that, $U_{i_0}$ generates $(s,a_{i_0}(x))$ and sends it to $U_{i_1}$. Then, an adversary can modify the authenticated message as $(s,a'_{i_0}(x))$, such that $a'_{i_0}(U_{i_1}) = a_{i_0}(U_{i_1})$ and $a'_{i_0}(U_{i_2}) \neq a_{i_0}(U_{i_2})$ for a certain user $U_{i_2}$. On receiving $(s,a'_{i_0}(x))$, $U_{i_1}$ accepts it as valid since $a'_{i_0}(U_{i_1}) = a_{i_1}(U_{i_0})$. However, when $U_{i_1}$ further transfers $(s,a'_{i_0}(x))$ to $U_{i_2}$, $U_{i_2}$ does not accept it since $a'_{i_0}(U_{i_2}) \neq a_{i_2}(U_{i_0})$, and $U_{i_1}$ will be suspected to have forged it. Following [32], we call this type of attack transfer with a trap. For this reason, DMRA (and MRA) cannot be used as a digital signature.

In the remaining part of this Section, we show an attack on the DMRA presented in Section 3.2.2, and also present a method to fix the problem. This attack is easy to perform and very effective. In this attack, by observing a valid authenticated message, $\omega - 1$ colluders can forge any user’s valid authenticated message with probability 1.
4.1.1 An attack on DMRA of Section 3.2.2

Let \( W = \{U_1, \ldots, U_{\omega-1}\} \) be the set of colluders. These colluders can forge one user’s authenticated message as follows. When \( U_0 (\not\in W) \) transmits a valid authenticated message \((s, a_0(x))\), the colluders interrupt it and use it for producing a forgery of another user’s authenticated message. On observing \((s, a_0(x))\), the colluders generate authenticated messages \((s, a_1(x)), (s, a_2(x)), \ldots, (s, a_{\omega-1}(x))\). Letting

\[
P_\ell(x, y) = \left( 1, x, x^2, \ldots, x^{\omega-1} \right) A_\ell \begin{pmatrix} 1 \\ y \\ y^2 \\ \vdots \\ y^{\omega-1} \end{pmatrix}
\]

for \( \ell = 0, 1 \), where \( A_\ell \) are \( \omega \times \omega \) symmetric matrices over \( F_q \), the colluders now have a matrix

\[
D = (A_0 + sA_1) \begin{pmatrix} 1 & 1 & \cdots & 1 \\ U_0 & U_1 & \cdots & U_{\omega-1} \\ \vdots & \vdots & \ddots & \vdots \\ U_0^{\omega-1} & U_1^{\omega-1} & \cdots & U_{\omega-1}^{\omega-1} \end{pmatrix}^{-1}
\]

Then, by using \( D \), the matrix \( A_0 + sA_1 \) can be easily obtained as follows:

\[
A_0 + sA_1 = D \begin{pmatrix} 1 & 1 & \cdots & 1 \\ U_0 & U_1 & \cdots & U_{\omega-1} \\ \vdots & \vdots & \ddots & \vdots \\ U_0^{\omega-1} & U_1^{\omega-1} & \cdots & U_{\omega-1}^{\omega-1} \end{pmatrix}^{-1}.
\]

If the colluders \( W \) want to forge an authenticated message of a user \( U_j \), where \( U_j (\not\in W \cup \{U_0\}) \), \( W \) calculate

\[
a'_j(x) = (1, U_j, U_j^2, \ldots, U_j^{\omega-1})(A_0 + sA_1),
\]

and broadcast \((s, a'_j(x))\) as an authenticated message of \( U_j \) for the source state \( s \). Since \((s, a'_j(x))\) is exactly equal to \( U_j \)'s valid authentication message for source state \( s \), the colluders succeed in impersonation (or entity substitution) for \( U_j \) with probability 1.
4.1.2 A Method to fix the problem

An essential problem in DMRA of Section 3.2.2 is that $A_0 + sA_1$ can be calculated by using both $\omega - 1$ colluders’ secret information and an authenticated message generated by a honest user. In order to avoid calculating $A_0 + sA_1$, the rank of $A_0 + sA_1$ must be larger than $\omega - 1$. This implies that the degree of $x$ and $y$ in $P_0(x, y)$ and $P_1(x, y)$ must be at least $\omega$. Letting the degree of $x$ and $y$ in $P_0(x, y)$ and $P_1(x, y)$ be at least $\omega$, the colluders cannot succeed in the above attack with non-negligible probability. It should be noted that both the required memory size for a user’s secret information and that for an authenticated message are increased by this modification.

4.2 Models of unconditionally secure signature

In this Section a model of unconditionally secure signature scheme with transferability is show. We assume that there is a Trusted Authority, denoted by TA, and $n$ users $U = \{U_1, U_2, \ldots, U_n\}$. For convenience we use for each user $U_i \in U$ ($1 \leq i \leq n$) the same symbol $U_i$ to denote the identity of the user. The TA produces a pair of signing and verification-keys that each user can use to generate and/or verify signatures respectively. A more formal definition is given below:

**Definition 4.1** A scheme $\Pi$ is an Unconditionally Secure Signature if it is constructed as follows:

1. Notation: $\Pi$ consists of $(TA, U, M, S, V, A, Sig, Ver)$ where

   - TA is a trusted authority,
   - $U$ is a finite set of users,
   - $M$ is a finite set of possible messages,
   - $S$ is a finite set of possible signing-keys,
   - $V$ is a finite set of possible verification-keys,
   - $A$ is a finite set of possible signatures,
   - Sig: $S \times M \rightarrow A$ is a signing algorithm,
   - Ver: $M \times A \times V \rightarrow \{\text{accept, reject}\}$ is a verification algorithm.
2. Key Pair Generation and Distribution by TA: For each user \( U_i \in \mathcal{U} \) \((1 \leq i \leq n)\), the TA chooses a signing key \( s_i \in \mathcal{S} \) and a verification-key \( v_i \in \mathcal{V} \), and transmits the pair \((s_i, v_i)\) to \( U_i \) via a secure channel. After delivering these keys, the TA may erase the pair \((s_i, v_i)\) from his memory and each user keeps his secret information secret.

3. Signature Generation: For a message \( m \in \mathcal{M} \), a user \( U_i \) generates a signature \( \alpha = \text{Sig}(s_i, m) \in \mathcal{A} \) by using his/her signing key in conjunction with the signing algorithm. The pair \((m, \alpha)\) is regarded as a signed message of \( U_i \).

4. Signature Verification: On receiving \((m, \alpha)\) from \( U_i \), the user \( U_j \) checks whether \( \alpha \) is valid by using his secret information \( e_j \). More precisely, \( U_j \) accepts \((m, \alpha)\) as a valid signed message from \( U_i \) if \( \text{Ver}(m, \alpha, e_j, U_i) = \text{accept} \).

4.2.1 Attacks against the schemes

Before discussing in a formal way the security of signature schemes in our model, we define the probability of success of various types of attacks. We consider three broad types of attacks: impersonation, substitution and transfer with a trap. Of these attacks, the first two are usually taken into account in discussing the security of authentication codes, especially \( A^2 \)-codes, \( A^3 \)-codes and MRA codes. The third type of attack, transfer with a trap, is new, and will be formally defined later.

Consider the case where there are \( n \) users among whom up to \( \omega - 1 \) users may be dishonest (and, hence, may collude). Each user is allowed to sign up to \( \psi \) signatures. We now discuss in a more formal way the three types of attacks.

1. Impersonation:

   \( t \) users, with \( t < \omega \), launch an attack against a pair of user \( U_i \) and \( U_j \) by generating a signed message with the hope that \( U_j \) accepts it as being a valid signature from \( U_i \). This attack may be executed after the colluders observe at most \( \psi(n - 1) \) signed messages generated by users other than \( U_i \).

2. Substitution:

   \( t \) users, with \( t < \omega \), construct a fraudulent message \( m' \) to replace a message genuinely signed by \( U_i \), with the hope that \( U_j \) will accept it as being an authentic message from \( U_i \). This attack may be executed after the colluders observe at most \( \psi n \) signed
messages generated by any users. Among the observed messages, at least one but up to $\psi$ may be generated by $U_i$.

3. Transfer with a trap:

After $U_j$ receives a valid pair $(m, \alpha)$ from $U_i$, $t$ colluders, where $t < \omega$, attempt to generate a new pair $(m, \alpha')$ with $\alpha \neq \alpha'$. Note that both the signer $U_i$ and the user $U_j$ could be among the colluders. The colluders hope that another user $U_k$ will accept $(m, \alpha')$ as being a valid message-signature pair from $U_i$, but no other users will. The risk with this attack is that when $U_j$ transfers such a pair $(m, \alpha')$ to $U_k$ and $U_k$ then transfers it to another user $U_l$, $U_l$ finds that the pair is invalid. When this happens, $U_k$ is in a sense trapped by the colluders.

To formally define the probabilities of success in the above three attacks, some notations are introduced. Let $W = \{W \subset U | |W| \leq \omega - 1\}$, where $\omega - 1$ is the maximum number of colluders among users. Each element of $W$ represents a group of possibly colluding users. Let $s_W = \{s_{k_1}, \ldots, s_{k_j}\}$ and $v_W = \{v_{k_1}, \ldots, v_{k_j}\}$, where $W = \{U_{k_1}, \ldots, U_{k_j}\}$ ($j \leq \omega - 1$), be the set of signing and verification keys for a $W \in W$. Each user is allowed to sign up to $\psi$ signatures.

**Definition 4.2** The success probabilities of impersonation, substitution and transfer with a trap attacks in a user signature scheme, denoted by $P_I$, $P_S$ and $P_T$ respectively, are formally defined as follows:

1. Success probability of impersonation: for $W \in W$ and $U_i, U_j \in U$ with $U_i, U_j \notin W$, we define $P_I(U_i, U_j, W)$ as:

$$P_I(U_i, U_j, W) = \max_{s_W, v_W} \max_{1 \leq k \leq n, k \neq i} \max_{c_k = \{(m_{k, \ell}, \alpha_{k, \ell})\}} \max_{(m, \alpha)} \Pr(U_j \text{ accepts } (m, \alpha) \text{ as valid from } U_i | s_W, v_W, \{c_k\}),$$

where $c_k = \{(m_{k, \ell}, \alpha_{k, \ell})\}$ is taken on a family of possible sets of valid user signed messages generated by a user $U_k$ ($1 \leq k \leq n, k \neq i$) such that $0 \leq |c_k| \leq \psi$ ($1 \leq k \leq n, k \neq i$). Note that $m_{k, \ell}$ are not necessarily distinct. Then, $P_I$ is given as $P_I = \max_{U_i, U_j, W} Pr(U_i, U_j, W)$, where $W \in W$ and $U_i, U_j \in U$ with $U_i, U_j \notin W$. 
2. Success probability of substitution: for \( W \in \mathcal{W} \) and \( U_i, U_j \in \mathcal{U} \) with \( U_i, U_j \notin W \), we define \( P_S(U_i, U_j, W) \) as

\[
P_S(U_i, U_j, W) = \max_{s_W, v_W} \max_{1 \leq k \leq n} \max_{c_k = \{(m_{k, \ell}, \alpha_{k, \ell})\}} \max_{(m, \alpha)} \Pr(U_j \text{ accepts } (m, \alpha) \text{ as valid from } U_i | s_W, v_W, \{c_k\}),
\]

where \( c_k = \{(m_{k, \ell}, \alpha_{k, \ell})\} \) is taken over a family of possible sets of valid signed messages generated by \( U_k \) (\( 1 \leq k \leq n \)) such that \( 0 < |c_i| \leq \psi \) and \( 0 \leq |c_k| \leq \psi \) (\( 1 \leq k \leq n, k \neq i \)) and \((m, \alpha)\) is taken such that \( m \neq m_{i, \ell} \) for any \( \ell \). Note that \( m_{k, \ell} \) are not necessarily distinct. Then, \( P_S \) is given as \( P_S = \max_{U_i, U_j, W} \Pr(U_i, U_j, W) \), where \( W \in \mathcal{W} \) and \( U_i, U_j \in \mathcal{U} \) with \( U_i, U_j \notin W \).

3. Success probability of transfer with a trap: for \( W \in \mathcal{W} \) and \( U_i, U_j \in \mathcal{U} \) with \( U_j \notin W \), we define \( P_T(U_i, U_j, W) \) as

\[
P_T(U_i, U_j, W) = \max_{s_W, v_W} \max_{1 \leq k \leq n, k \neq i} \max_{c_k = \{(m_{k, \ell}, \alpha_{k, \ell})\}} \max_{(m, \alpha), \alpha'} \Pr(U_j \text{ accepts } (m, \alpha') \text{ as valid from } U_i | s_W, v_W, (m, \alpha)),
\]

where \( c_k = \{(m_{k, \ell}, \alpha_{k, \ell})\} \) is taken over a family of possible sets of valid signed messages generated by \( U_k \) (\( 1 \leq k \leq n, k \neq i \)) such that \( 0 \leq |c_k| \leq \psi \) (\( 1 \leq k \leq n, k \neq i \)), \((m, \alpha)\) is taken over the set of possible signed message generated by \( U_i \) and \( \alpha' \) is taken such that \( \alpha \neq \alpha' \). Then \( P_T \) is given as \( P_T = \max_{U_i, U_j, W} \Pr(U_i, U_j, W) \), where \( W \in \mathcal{W} \) and \( U_i, U_j \in \mathcal{U} \) with \( U_j \notin W \).

Now we are ready to define the concept of an \((n, \omega, \psi, p_1, p_2)\) secure ISSUSG signature scheme. Here both \( p_1 \) and \( p_2 \) are security parameters whose meanings will be made precise in the following definition.

**Definition 4.3** Let \( \Pi \) be an ISSUSG with \( n \) users. Then, \( \Pi \) is \((n, \omega, \psi, p_1, p_2)\) secure if the following conditions are satisfied: as long as there exist at most \( \omega \) colluders and each user is allowed to generate at most \( \psi \) signatures, the following inequalities hold:

\[
\max\{P_I, P_S\} \leq p_1
\]

\[
P_T \leq p_2
\]

where \( P_I \), \( P_S \) and \( P_T \) are the probabilities of success in impersonation, substitution and transfer with a trap attacks, respectively.
We note that there is an alternative definition of security in which one may use a single security parameter $p$ and define the success probability as
\[ \max\{P_I, P_S, P_T\} \leq p. \]

In practice, however, some applications may be interested in enforcing security against impersonation and substitution and not against transfer with a trap, while some other applications may have an emphasis on robustness against transfer with a trap. By introducing two separate parameters $p_1$ and $p_2$, we have the opportunity of designing a signature scheme with fine-tuned level of security.

### 4.2.2 Two constructions of unconditionally secure signature schemes

In this Section, we show two constructions of unconditionally secure digital signature schemes, a symmetric construction and an asymmetric one. In these schemes, even if the flexibility of the parameters is partially lost, the required memory sizes are reduced considerably compared to the previous method. More precisely, in the next schemes, the number of signature users can generate is determined to be only one and then $\psi = 1$.

**A symmetric construction**

In this Subsection, we show an implementation of a One-Time ISSUSG in which $S = V$. In this construction, memory size of user’s secret information is partially reduced. The construction is based on the same ideas of the scheme presented in Section 3.2.2. Namely, we introduce symmetric functions for unifying the secret information for signing and for verifying. However, it should be noted that it is not trivial to implement, since the scheme presented in Section 3.2.2 does not fulfill the transferability property. The essential reason for which it does not provide transferability is that, for $U_i$’s authenticated message $(s_i, a_i(x))$, any entity can calculate $a_i(U_j)$ and find another function $a_i(x)'$ such that $a_i(x) \neq a_i(x)'$ and $a_i(U_j)' = a_i(U_j)$. This is hard to solve since $U_j$ must be public. In the following we indicate with $E$ the set of secret information.

1. **Key Generation and Distribution by TA**: Let $F_{q_0}$ be a finite field with $q_0$ elements such that $q_0 \geq n \omega q$, where $q$ is a security parameter of the system. We assume that the size of $q_0$ is almost the same as $n \omega q$. Then TA divides $F_{q_0}$ into $n$ disjoint
subsets $U_1, \ldots, U_n$, such that $|U_i| = \omega q$ for every $i$, and $U_i \cap U_j = \emptyset$ if $i \neq j$. Here $U_i$ $(1 \leq i \leq n)$ are made public for all users. For each user $U_i$ $(1 \leq i \leq n)$, the TA picks uniformly at random a number $u_i$ from $U_i$, respectively, and chooses uniformly at random two symmetric polynomials $P_0(x, y)$ and $P_1(x, y)$ over $F_{q^0}$ with two variables $x$ and $y$ of degree at most $\omega$. Moreover, we assume that each message $m$ is an element in $F_{q^0}$ as well. For each user $U_i$ $(1 \leq i \leq n)$, the TA computes his secret information

$$e_i = \{P_0(x, u_i), P_1(x, u_i), u_i\}.$$ 

Then, the TA sends $e_i$ to $U_i$ over a secure channel. Once the secret information has been delivered, there is now no need for the TA to keep user’s secret information.

2. **Signature Generation:** For a message $m \in F_{q^0}$, $U_i$ generates a signature by $\alpha = \{a_{i,m}(x), u_i\}$ using his secret information, where $a_{i,m}(x) = P_0(x, u_i) + mP_1(x, u_i)$. Then, $(m, \alpha)$ is sent by $U_i$ with his identity $U_i$.

3. **Signature Verification:** On receiving $U_i$’s signature $(m, \alpha)$, user $U_j$ checks whether $\alpha$ is valid or not, by using his secret information $e_j$. More precisely, $U_j$ accepts $(m, \alpha)$ as being a valid message-signature pair from $U_i$ if

$$(P_0(x, u_j) + mP_1(x, u_j))|_{x=u_i} = a_{i,m}(x)|_{x=u_j}$$

and $u_i \in U_i$.

**Theorem 4.1** The above scheme results in an $(n, \omega, \frac{1}{q^0}, \frac{1}{q})$-secure One-Time ISSUSG scheme.

**Proof:** Assume that after seeing a signed message $(m_{i_0}, \alpha)$ published by $U_{i_0}$, the colluders $U_1, \ldots, U_{\omega-1}$ want to generate $(m_{i_1}, \alpha')$, such that $m_{i_1} = m_{i_0}$ and the user $U_{i_2}$ will accept it as a valid signed message of the user $U_{i_1}$, i.e. $\alpha'$ consists of $\{u'_{i_1}, a_{i_1,m_{i_1}}(x)\}$ such that $a_{i_1,m_{i_1}(u_{i_2})} = P_0(u'_{i_1}, u_{i_2}) + mP_1(u'_{i_1}, u_{i_2})$ and $u'_{i_1} \in U_{i_1}$. Letting

$$P_\ell(x, y) = \left(1, x, x^2, \ldots, x^{\omega}\right) A_{\ell} \begin{pmatrix} 1 \\ y \\ y^2 \\ \vdots \\ y^\omega \end{pmatrix}$$


for $\ell = 0, 1$ where $A_{\ell}$ are a $(\omega + 1) \times (\omega + 1)$ symmetric matrices over $F_{q_0}$, the colluders have a $(\omega + 1) \times \omega$ matrix $D$, where

\[
D = (A_0 + m_{i_0}A_1) \begin{pmatrix}
1 & 1 & \ldots & 1 \\
U_{i_0} & U_1 & \ldots & U_{\omega-1} \\
U_{i_0}^2 & U_1^2 & \ldots & U_{\omega-1}^2 \\
\vdots & \vdots & \ddots & \vdots \\
U_{i_0}^\omega & U_1^\omega & \ldots & U_{\omega-1}^\omega
\end{pmatrix}.
\]

From Lemma 3.1, there exist $q_0$ different matrices $X$ such that

\[
D = X \begin{pmatrix}
1 & 1 & \ldots & 1 \\
U_{i_0} & U_1 & \ldots & U_{\omega-1} \\
U_{i_0}^2 & U_1^2 & \ldots & U_{\omega-1}^2 \\
\vdots & \vdots & \ddots & \vdots \\
U_{i_0}^\omega & U_1^\omega & \ldots & U_{\omega-1}^\omega
\end{pmatrix}.
\]

This implies that there are $q_0$ different values for $A_0 + m_{i_0}A_1$.

In order for the colluders to succeed the attack, they need to find a pair of $u'_{i_1}$ and $a'_{i_1,m_{i_1}}(x)$ such that

\[
a'_{i_1,m_{i_1}}(u_{i_2}) = \left(1, u'_{i_1}, u'_{i_1}^2, \ldots, u'_{i_1}^\omega\right) (A_0 + m_{i_0}A_1) \begin{pmatrix}
1 \\
u_{i_2} \\
u_{i_2}^2 \\
\vdots \\
u_{i_2}^\omega
\end{pmatrix}
\]

and $u'_{i_1} \in U_{i_1}$. Letting $d$ be

\[
\left(1, u'_{i_1}, u'_{i_1}^2, \ldots, u'_{i_1}^\omega\right) (A_0 + m_{i_0}A_1) \begin{pmatrix}
1 \\
u_{i_2} \\
u_{i_2}^2 \\
\vdots \\
u_{i_2}^\omega
\end{pmatrix}
\]

$q_0$ different matrices for $A_0 + m_{i_0}A_1$ result in $q_0$ different values for $d$. This indicates that the probability of succeeding in finding $a'_{i_1,m_{i_1}}(x)$, such that $a'_{i_1,m_{i_1}}(u_{i_2}) = d$, does not exceed $\frac{1}{q_0}$, i.e. $P_I = \frac{1}{q_0}$. Similarly, we can prove $P_S \leq \frac{1}{q_0}$ and $P_T = \frac{1}{q}$. 

Here, we briefly show the proof for $P_T = \frac{1}{q}$. Assume that after seeing a signed message $(m_{i_0}, \alpha)$ published by $U_{i_0}$, the colluders $U_1, \ldots, U_{\omega-1}$ want to generate $(m_{i_0}, \alpha')$, such that $\alpha' \neq \alpha$ and the user $U_{i_1}$ will accepts it as a valid signed message of the user $U_{i_0}$. Let $\alpha$ be $\{u_{i_0}, a_{i_0, m_{i_0}}(x)\}$ as described before. Since $a_{i_0, m_{i_0}}(x)$ is a polynomial with a variable $x$ of degree at most $\omega$, $a_{i_0, m_{i_0}}(x)' (a_{i_0, m_{i_0}}(x)' \neq a_{i_0, m_{i_0}}(x))$ has at most $\omega$ pairs of $\{c, a_{i_0, m_{i_0}}(c)\}$, such that $c \in F_q$ and $a_{i_0, m_{i_0}}(c)' = a_{i_0, m_{i_0}}(c)$, where $a_{i_0, m_{i_0}}(x)'$ is a polynomial with a variable $x$ of degree at most $\omega$. Hence, the best strategy for succeeding in transfer with a trap is as follows: The colluders choose uniformly at random $\omega$ distinct numbers $u_{i_1}^{(1)}, \ldots, u_{i_1}^{(\omega)}$ from $U_{i_1}$ and generate $a'_{i_0, m_{i_0}}(x)$ ($a'_{i_0, m_{i_0}}(x) \neq a_{i_0, m_{i_0}}(x)$) such that $a'_{i_0, m_{i_0}}(u_{i_1}^{(1)}) = a_{i_0, m_{i_0}}(u_{i_1}^{(1)}), a'_{i_0, m_{i_0}}(u_{i_1}^{(2)}) = a_{i_0, m_{i_0}}(u_{i_1}^{(2)}), \ldots, a'_{i_0, m_{i_0}}(u_{i_1}^{(\omega)}) = a_{i_0, m_{i_0}}(u_{i_1}^{(\omega)})$. Then, the colluders send $\alpha' = \{u_{i_0}, a'_{i_0, m_{i_0}}(x)\}$ to $U_{i_1}$. The attack is successful if and only if $u_{i_1} \in \{u_{i_1}^{(1)}, u_{i_1}^{(2)}, \ldots, u_{i_1}^{(\omega)}\}$. Hence, $P_T = \frac{\omega^2}{q^2} = \frac{1}{q}$.

\[|\mathcal{A}| = \omega q^2 + 1 \quad \text{(size of signature)}\]
\[|\mathcal{E}| = \omega q^2 + 2 \quad \text{(size of secret information)}\]

Although in this scheme the required memory size of a signature is slightly increased compared to [32], user’ secret information is significantly reduced.

**An asymmetric construction**

In this Subsection, we show other methods for reducing user memory size without increasing the size for a signature. One of the proposed schemes in this Subsection is optimal, especially in term of the size of a signature. Such schemes are called asymmetric since the secret information for signing and verifying are different, i.e. $S \neq V$. In the following we indicate with $\mathcal{E}$ the set of secret information.

1. **Key Pair Generation and Distribution by TA:** Let $F_q$ be a finite field with $q$ elements such that $q \geq n$. The TA picks $n$ elements $v_1, v_2, \ldots, v_n$ uniformly at random in $F_q^{\omega-1}$...
Chapter 4. Unconditionally secure signatures

47

for the users $U_1, U_2, \ldots, U_n$ respectively, and chooses two polynomials uniformly
at random, $G_0(x,y_1,\ldots,y_{\omega-1})$ and $G_1(x,y_1,\ldots,y_{\omega-1})$, over $F_q$ with $\omega$
variables $x, y_1, \ldots, y_{\omega-1}$, in which the degree of $x$ is at most $\omega$ and that of every $y_i$
is at most 1. Moreover, we assume that each user’s identity $U_i$ and message $m$ are elements of
$F_q$. For each user $U_i$ ($1 \leq i \leq n$), the TA computes $U_i$’s secret information

$$e_i = \{G_0(U_i,y_1,\ldots,y_n), G_1(U_i,y_1,\ldots,y_n), G_0(x,v_i), G_1(x,v_i), v_i\}.$$ 

The TA sends $e_i$ to $U_i$ over a secure channel. Once all the keys are delivered, there
is no need for the TA to keep user’s secret information.

2. Signature Generation: For a message $m \in F_q$, $U_i$ generates a signature by $\alpha =
G_0(U_i,y_1,\ldots,y_{\omega-1}) + mG_1(U_i,y_1,\ldots,y_{\omega-1})$ using $G_0(U_i,y_1,\ldots,y_{\omega-1})$ and $G_1(U_i,y_1,\ldots,y_{\omega-1})$. Then, $(m, \alpha)$ is sent by $U_i$ with his identity $U_i$.

3. Signature Verification: On receiving $(m, \alpha)$ from $U_i$, user $U_j$ checks whether $\alpha$ is
valid by using his secret information. More precisely, $U_j$ accepts $(m, \alpha)$ as being a
valid message-signature pair from $U_i$ if

$$(G_0(x,v_j) + mG_1(x,v_j))|_{x=U_i} = \alpha|_{(y_1,\ldots,y_{\omega-1})=(v_1,j,\ldots,v_{\omega-1},j)}.$$ 

Theorem 4.3 The above scheme results in an $(n, \omega, (2^{-\frac{1}{q}})^{\frac{1}{2}}, \frac{1}{q})$-secure One-Time ISSUSG
scheme.

Proof: The proof of this theorem is similar to that of Theorem 4.1. \hfill \blacksquare

The above scheme can be slightly modified, resulting in another $(n, \omega, (\frac{1}{q^2}^{-\frac{1}{q}}, \frac{1}{q})$-secure
One-Time ISSUSG scheme.

Theorem 4.4 In the above construction, the following modification also produces an
$(n, \omega, (\frac{1}{q^2}, \frac{1}{q}))$-secure One-Time ISSUSG scheme: Instead of choosing randomly, the TA
may choose $n$ elements $v_1, \ldots, v_n \in F_{q^\omega-1}$, for user’s secret, such that for any $\omega$ vectors

$$v_{\omega_1} = (v_{1,i_1}, \ldots, v_{\omega-1,i_1}), \ldots, v_\omega = (v_{1,i_\omega}, \ldots, v_{\omega-1,i_\omega}),$$

the $\omega$ new vectors $(1, v_{1,i_1}, \ldots, v_{\omega-1,i_1}), \ldots, (1, v_{1,i_\omega}, \ldots, v_{\omega-1,i_\omega})$ are linearly independent.
Though the proposed \((n, \omega, \frac{1}{q}, \frac{1}{q})\)-secure One Time ISSUSG scheme is more secure than the proposed \((n, \omega, \frac{2}{q} - \frac{1}{q}, \frac{1}{q})\)-secure One-Time ISSUSG scheme in terms of impersonation or substitution, it requires more complicated transactions for generating each user’s secret information.

**Theorem 4.5** In the above constructions we have that:

\[ |A| = q^\omega \quad \text{(size of a signature)} \]
\[ |\mathcal{E}| = q^{5\omega + 1} \quad \text{(size of a user’s secret information).} \]

**Corollary 4.1** The construction proposed in Theorem 4.4 is optimal in terms of size of a signature.

Since the model of One-Time ISSUSG is regarded as a restricted version of that of MRA, lower bounds on required memory sizes for MRA can also be applied to One-Time ISSUSG. The required memory size for the above construction matches the lower bound on signature presented in Theorem 3.2.

### 4.2.3 Multiple messages signature scheme

In this Section we present an implementation of a DMRA scheme with transferability for multiple messages. It is constructed by using a polynomial with \(\omega + 1\) variables over a finite field.

As before, let \(\mathcal{U} = \{U_1, U_2, \ldots, U_n\}\) be the set of \(n\) users and TA be the Trusted Authority.

1. **Key Pair Generation and Distribution by TA:** Let \(F_q\) be a finite field with \(q\) elements such that \(q \geq n\). The TA picks uniformly at random \(n\) elements \(v_1, v_2, \ldots, v_n\) in \(F_q^{\omega - 1}\) for users \(U_1, U_2, \ldots, U_n\) respectively and constructs a polynomial \(F(x, y_1, \ldots, y_{\omega - 1}, z)\) as follows:

\[
F(x, y_1, \ldots, y_{\omega - 1}, z) = \sum_{i=0}^{\omega} \sum_{k=0}^{\psi} a_{0ik}x^iz^k + \sum_{i=0}^{\omega-1} \sum_{j=1}^{\psi} \sum_{k=0}^{\omega-1} a_{ijk}x^iy_jz^k
\]
where the coefficients $a_{ijk}$ are chosen uniformly at random from $F_q$. Moreover, we assume that each user’s identity $U_{\ell}$ and message $m$ are also from $F_q$.

For each user $U_{\ell}$ ($1 \leq \ell \leq n$), the TA computes a signing key $s_{\ell} = F(U_{\ell}, y_1, \ldots, y_{\omega-1}, z)$ and a verification key $\tilde{v}_{\ell} = F(x, v_{\ell}, z)$. $v_{\ell}$ and $\tilde{v}_{\ell}$ together form a pair of verification-keys for user $U_{\ell}$. The TA then sends both the signing-key and the pair of verification-keys to $U_{\ell}$ over a secure channel. Once the keys are delivered, there is no need for the TA to keep user’s keys.

2. **Signature Generation**: For a message $m \in F_q$, $U_i$ generates a signature by

$$\alpha = F(U_i, y_1, \ldots, y_{\omega-1}, z) \mid z = m = F(U_i, y_1, \ldots, y_{\omega-1}, m)$$

using his signing-key.

3. **Signature Verification**: On receiving the pair $(m, \alpha)$ from $U_i$, user $U_j$ checks whether $\alpha$ is valid by using of his verification keys $v_j$ and $\tilde{v}_j$. More specifically, $U_j$ calculates values $r_1$ and $r_2$ using his verification keys $\tilde{v}_j = F(x, v_j, z)$ and $v_j = (v_{1,j}, \ldots, v_{\omega-1,j})$ as follows:

$$r_1 = F(x, v_j, z) \mid x = U_i, z = m,$$

$$r_2 = \alpha \mid y_1, \ldots, y_{\omega-1} = (v_{1,j}, \ldots, v_{\omega-1,j}).$$

$U_j$ accepts $(m, \alpha)$ as being a valid message-signature pair from $U_i$ if and only if $r_1 = r_2$.

We can show that the above signature scheme is an $(n, \omega, \psi, (2 - \frac{1}{q^2}), \frac{1}{q})$-secure scheme.

**Theorem 4.6** The above scheme results in an $(n, \omega, \psi, (2 - \frac{1}{q^2}), \frac{1}{q})$-secure scheme.

**Proof:** The proof of this theorem can be obtained with the same lines of Theorem 4.1.

**Theorem 4.7** In the above construction, the following modification also produces an $(n, \omega, \frac{1}{q}, \frac{1}{q-1})$-secure ISSUSG scheme: Instead of choosing randomly, the TA may choose $n$ elements $v_1, \ldots, v_n \in F_q^{\omega-1}$, for user’s secret, such that for any $\omega$ vectors

$$v_{i_1} = (v_{1,i_1}, \ldots, v_{\omega-1,i_1}), \ldots, v_{i_\omega} = (v_{1,i_\omega}, \ldots, v_{\omega-1,i_\omega}),$$
the $\omega$ new vectors $(1, v_{1,i_1}, \ldots, v_{\omega-1,i_1}), \ldots, (1, v_{1,i_\omega}, \ldots, v_{\omega-1,i_\omega})$ are linearly independent.

Note that this scheme can be used in place of an authentication code, MRA or DMRA. In fact this scheme is cryptographically stronger than the authentication codes, with an added benefit of being transferable, although it requires more memory space than MRA and DMRA.

Due to Theorem 4.7, the above scheme can be slightly modified, resulting in another $(n, \omega, \psi, \frac{1}{q}, \frac{1}{q-1})$.

The resources required by our construction are quantified in the following Theorem.

**Theorem 4.8** In the above constructions we have that:

$$|A| = q^\omega, \ (\text{size of signature})$$
$$|S| = q^{\omega(\psi+1)}, \ (\text{size of signing-key})$$
$$|V| = q^{\omega+n(\psi+1)}, \ (\text{size of verification-key}).$$

**Corollary 4.2** The construction proposed in Theorem 4.7 is optimal in term of the size of a signature.

This Corollary follows because the required memory size for the above construction matches the lower bound on a signature presented in Theorem 3.2.
Chapter 5

Unconditionally Secure Group Signatures

5.1 What are Unconditionally Secure Group Signatures?

Unconditionally Secure Group Signature Schemes are an extension of One-Time ISSUSG schemes presented in [33] and described in Chapter 4. A group signature scheme allows a group member to sign messages on behalf of the group. In case of a dispute, the identity of the signature’s originator can be revealed by a Trusted Authority or with the help of the signer. In the following we introduce two models of Unconditionally Secure Group Signature: in one the group identity is opened by the TA and in the second it is opened with the help of the signer.

5.2 Unconditionally Secure Group Signature opened by TA

5.2.1 The model

In this Subsection a model of Unconditionally Secure Group Signature schemes opened by TA is described in which each user can generate and verify two types of signature: user signature and group signature. User signatures are generated by a user that wants to reveals his identity, while with group signatures a user doesn’t reveal his identity, but only that he is a member of a group. Both types of signatures are assumed to work in a group. New users are allowed to join the group even after the system is set up, as long as the total number of users does not exceed a pre-defined threshold (this threshold is denoted by \( n \)). When the threshold is sufficiently large, our signature schemes can be used in many applications when conventional public key signature schemes are used.
Therefore, the group orientation of our schemes should not present any difficulties in practical applications.

We assume that there is a Trusted Authority, denoted by TA, and \( n \) users \( \mathcal{U} = \{U_1, U_2, \ldots, U_n\} \). For convenience we use for each user \( U_i \in \mathcal{U} (1 \leq i \leq n) \) the same symbol \( U_i \) to denote the identity of the user, and with \( U_{G,i} \) the identity used by user \( U_i \) to generate a group signature. The TA produces a pair of signing and verification-keys on behalf of the users. Once being given a pair of keys, any user can generate and/or verify signatures (user or group signatures) by using signing-key and verification-key, respectively. A more formal definition is given below:

**Definition 5.1** A scheme \( \Pi \) is an Unconditionally Secure Group Signature opened by TA if it is constructed as follows:

1. **Notation:** \( \Pi \) consists of \((\text{TA}, \mathcal{U}, \mathcal{M}, \mathcal{S}, \mathcal{V}, \mathcal{A}, \text{Sig}, \text{Ver}, \text{Open})\) where
   - \( \text{TA} \) is a trusted authority,
   - \( \mathcal{U} \) is a finite set of users,
   - \( \mathcal{M} \) is a finite set of possible messages,
   - \( \mathcal{S} \) is a finite set of possible signing-keys,
   - \( \mathcal{V} \) is a finite set of possible verification-keys,
   - \( \mathcal{A} \) is a finite set of possible signatures,
   - \( \text{Sig} : \mathcal{S} \times \mathcal{M} \rightarrow \mathcal{A} \) is a signing algorithm,
   - \( \text{Ver} : \mathcal{M} \times \mathcal{A} \times \mathcal{V} \times \{\mathcal{U} \cup \emptyset\} \rightarrow \{\text{accept, reject}\} \) is a verification algorithm,
   - \( \text{Open} : \mathcal{A} \rightarrow \mathcal{U} \) is an algorithm that TA can use to compute signer’s identity from user’s group identity.

2. **Key Pair Generation and Distribution by TA:** For each user \( U_i \in \mathcal{U} (1 \leq i \leq n) \), the TA chooses a signing key \( s_i \in \mathcal{S} \) and a verification-key \( v_i \in \mathcal{V} \), and transmits the pair \((s_i, v_i)\) to \( U_i \) via a secure channel. After delivering these keys, the TA may erase the pair \((s_i, v_i)\) from his memory and each user keeps his secret information secret.

3. **Signature Generation:** For a message \( m \in \mathcal{M} \), a user \( U_i \) generate a signature \( \alpha = \text{Sig}(s_i, m) \in \mathcal{A} \) by using his/her signing key in conjunction with the signing algorithm. The pair \((m, \alpha)\) is regarded as:
• a signed message of \(U_i\),
• a group signed message that is a signature generated by a member of the group using \(U_{G,i}\) as his identity.

4. **Signature Verification**: On receiving \((m, \alpha)\), the user \(U_j\) checks if the signature is a user signature or a group signature. If \((m, \alpha)\) is a user signature then \(U_j\) accepts it as valid from \(U_i\) if \(\text{Ver}(m, \alpha, v_j, U_i) = \text{accept}\), else if \((m, \alpha)\) is a group signature then \(U_j\) accept it as valid if \(\text{Ver}(m, \alpha, v_j, U_{G,i}) = \text{accept}\). Of course if a member of a group knows \(U_{G,i}\), he cannot determine the identity of the signer.

5. **Open a signature**: On receiving \(U_{G,i}\) from a user \(U_j\), TA can compute \(U_i\) that is the identity of a user that has generated a group signature.

The main difference between the above definition and the previous one of Section 4.2 is in the introduction of a new type of signature, referred to as **group signature**.

Before describing the proposed constructions, we define a new operation.

**Definition 5.2** Given a polynomial \(P(x_1, x_2, \ldots, x_\ell) = \sum_{i_1=0}^{k_1} \sum_{i_2=0}^{k_2} \cdots \sum_{i_\ell=0}^{k_\ell} a_{i_1, \ldots, i_\ell} x_1^{i_1} x_2^{i_2} \cdots x_\ell^{i_\ell}\) and a vector \(C = (c_1, c_2, \ldots, c_\ell)\) we define the operator \(\oplus\) as:

\[
P(x_1, x_2, \ldots, x_\ell) \oplus C = \sum_{i_1=0}^{k_1} \sum_{i_2=0}^{k_2} \cdots \sum_{i_\ell=0}^{k_\ell} a_{i_1, \ldots, i_\ell} (x_1 c_1)^{i_1} (x_2 c_2)^{i_2} \cdots (x_\ell c_\ell)^{i_\ell}.
\]

5.2.2 **Properties and attacks**

In order to discuss in a formal way the security of Unconditionally Secure Group Signature Schemes, we define some properties and the probability of success of various types of attacks. We consider two properties for group signature: **Full-Anonymity** and **Full-Traceability**, and three broad types of attacks: **Impersonation**, **Substitution** and **Transfer with a trap**. We don’t describe the above attacks because we have already seen them in Section 4.2.1.

For group signatures there are two new properties to consider referred to as Full-Anonymity and Full-Traceability. They can be informally defined as follows:

1. **Full-Anonymity**: This property requires that an adversary who does not possess the group manager’s secret key finds it hard to recover the identity of the signer from its group signature;
2. **Full-Traceability:** It requires that no colluding set of group members can create signatures that cannot be opened or signatures that cannot be traced back to members of the coalition.

Before formally defining both the success probability of impersonation, substitution and transfer with a trap attacks and the probability of obtaining Full-Anonymity and Full-Traceability, some notations are introduced first. Let $W = \{ W \subset U : |W| \leq \omega - 1 \}$, where $\omega - 1$ is the maximum number of colluders among users. Each element of $W$ represents a group of possibly colluding users. Let $s_W = \{ s_{k_1}, \ldots, s_{k_j} \}$ and $v_W = \{ v_{k_1}, \ldots, v_{k_j} \}$, where $W = \{ U_{k_1}, \ldots, U_{k_j} \}$ ($j \leq \omega - 1$), be the set of signing and verification keys for a $W \in W$.

Definition 4.2 give us a formal way to define the success probability of impersonation, substitution and transfer with a trap attacks in a signature scheme, i.e. $P_I$, $P_S$ and $P_T$.

In the following, we extend this definition to group signature.

**Definition 5.3** The success probabilities of impersonation, substitution and transfer with a trap attacks in a group signature scheme opened by TA, denoted by $P^TA_{G,I}$, $P^TA_{G,S}$ and $P^TA_{G,T}$ respectively, are formally defined as follows:

1. **Success probability of Impersonation:** for $W \in W$ and $U_i, U_j \in U$ with $U_i, U_j \not\in W$, we define $P^TA_{G,I}(U_i, U_j, W)$ as:

$$P^TA_{G,I}(U_i, U_j, W) = \max_{s_W, v_W} \max_{1 \leq k \leq n, k \neq i} \max_{c_k = \{(m_{k,\ell}, \alpha_{k,\ell})\}} \max_{(m, \alpha)} \max_{u_{G,i} \in U} \Pr(TA \text{ traced } u_{G,i} \text{ to user } U_i) \Pr(U_j \text{ accepts } (m, \alpha) \text{ as a valid group signature}|s_W, v_W, \{c_k\})$$

where $u_{G,i}$ is the group identity of $U_i$, $c_k = \{(m_{k,\ell}, \alpha_{k,\ell})\}$ is taken over a family of possible sets of valid user signed messages generated by a user $U_k$ ($1 \leq k \leq n, k \neq i$) such that $0 \leq |c_k| \leq \psi$ ($1 \leq k \leq n, k \neq i$). Note that $m_{k,\ell}$ are not necessarily distinct. Then, $P^TA_{G,I}$ is given as $P^TA_{G,I} = \max_{U_i, U_j, W} P^TA_{G,I}(U_i, U_j, W)$, where $W \in W$ and $U_i, U_j \in U$ with $U_i, U_j \not\in W$.

2. **Success probability of Substitution:** for $W \in W$ and $U_i, U_j \in U$ with $U_i, U_j \not\in W$, we define $P^TA_{G,S}(U_i, U_j, W)$ as

$$P^TA_{G,S}(U_i, U_j, W) = \max_{s_W, v_W} \max_{1 \leq k \leq n} \max_{c_k = \{(m_{k,\ell}, \alpha_{k,\ell})\}} \max_{(m, \alpha)} \max_{u_{G,i} \in U}$$
Chapter 5. Unconditionally Secure Group Signatures

55

Traceability in a group signature scheme opened by TA, denoted by $P$

Definition 5.4

The success probabilities of an attack against Full-Anonymity and Full-

Traceability in a group signature scheme opened by TA, denoted by $P^{TA}_{FA}$ and $P^{TA}_{FT}$ respectively, are formally defined as follows:

1. Probability of Full-Anonymity: for $W \in W$ and $U_i \in U$ with $U_i \notin W$, we define $P^{TA}_{FA}(W, U_i)$ as:

$$P^{TA}_{FA}(W, U_i) = \max_{s_W \in W, \ell \leq n} \max_{c_k = \{ (m_k, \ell, \alpha_{k, \ell}) \}} \max_{GId = \{ u_{G,i} \}} \max_{\alpha' \in \mathcal{A}} \Pr(W \text{ identifies } U_i | u_{G,i}, s_W, v_W, \{ c_k \}, GId),$$

where $u_{G,i}$ is the group identity of user $U_i$, $c_k = \{ (m_k, \ell, \alpha_{k, \ell}) \}$ is taken over a family of possible sets of valid signed messages generated by $U_k$ ($1 \leq k \leq n$) such that $0 < |c_k| \leq \psi$ and $0 \leq |c_k| \leq \psi$ ($1 \leq k \leq n, k \neq i$) and $(m, \alpha)$ is such that $m \neq m_{i, \ell}$ for any $\ell$. Note that $m_{i, \ell}$ are not necessarily distinct. Then, $P^{TA}_{G,S}$ is given as $P^{TA}_{G,S} = \max_{U_i, U_j, W} P^{TA}_{G,S}(U_i, U_j, W)$, where $W \in W$ and $U_i, U_j \in U$ with $U_i, U_j \notin W$.

3. Success probability of Transfer with a trap: for $W \in W$ and $U_i, U_j \in U$ with $U_j \notin W$ we define $P^{TA}_{G,T}(U_i, U_j, W)$ as

$$P^{TA}_{G,T}(U_i, U_j, W) = \max_{s_W, v_W, \ell \leq n} \max_{c_k = \{ (m_k, \ell, \alpha_{k, \ell}) \}} \max_{(m, \alpha) \in \mathcal{A}} \max_{u_{G,i} \in U} \Pr(TA \text{ traced } u_{G,i} \text{ to user } U_i) \cdot \Pr(U_j \text{ accepts } (m, \alpha') \text{ as a valid group signature } |s_W, v_W, (m, \alpha)),$$

where $u_{G,i}$ is the group identity of $U_i$, $c_k = \{ (m_k, \ell, \alpha_{k, \ell}) \}$ is taken over a family of possible sets of valid signed messages generated by $U_k$ ($1 \leq k \leq n, k \neq i$) such that $0 \leq |c_k| \leq \psi$ ($1 \leq k \leq n, k \neq i$), $(m, \alpha)$ is taken over a the set of possible signed messages generated by $U_i$ and $\alpha'$ is such that $\alpha \neq \alpha'$. Then $P^{TA}_{G,T}$ is given as $P^{TA}_{G,T} = \max_{U_i, U_j, W} P^{TA}_{G,T}(U_i, U_j, W)$ where $W \in W$ and $U_i, U_j \in U$ with $U_j \notin W$.

Definition 5.4 The success probabilities of an attack against Full-Anonymity and Full-
Traceability in a group signature scheme opened by TA, denoted by $P^{TA}_{FA}$ and $P^{TA}_{FT}$ respectively, are formally defined as follows:

1. Probability of Full-Anonymity: for $W \in W$ and $U_i \in U$ with $U_i \notin W$, we define $P^{TA}_{FA}(W, U_i)$ as:

$$P^{TA}_{FA}(W, U_i) = \max_{s_W \in W, \ell \leq n} \max_{c_k = \{ (m_k, \ell, \alpha_{k, \ell}) \}} \max_{GId = \{ u_{G,i} \}} \max_{\alpha' \in \mathcal{A}} \Pr(W \text{ identifies } U_i | u_{G,i}, s_W, v_W, \{ c_k \}, GId),$$

where $u_{G,i}$ is the group identity of user $U_i$, $c_k = \{ (m_k, \ell, \alpha_{k, \ell}) \}$ is taken over a family of possible sets of valid signed messages generated by $U_k$ ($1 \leq k \leq n$) such that
Chapter 5. Unconditionally Secure Group Signatures

0 ≤ |c_k| ≤ ψ (1 ≤ k ≤ n) and GId is a set of group identities know by W. Then, \( P_{FT}^{TA} \) is given as
\[ P_{FT}^{TA} = \max_{W \in W} P_{FT}(W, U_i). \]

2. Probability of Full-Traceability: for \( W \in W \) and \( U_i \in U \) with \( U_i \not\in W \), we define
\[ P_{FT}^{TA}(U_i, W) = \max_{s_W, v_W} \max_{1 ≤ k ≤ n, k \neq i} \max_{c_k = \{(m_k, \alpha_k, \ell_k)\}} \max_{GId = \{u_{G,k'}\}} \Pr(U_j \text{ accepts } (m, \alpha) \text{ as a valid group signature} | s_W, v_W, \{c_k\}, GId) \]
\[ \Pr(TA \text{ cannot trace back to any member of } U), \]
where GId is a set of group identities know by W and \( c_k = \{(m_k, \alpha_k, \ell_k)\} \) is taken over a family of possible sets of valid user signed messages generated by a user \( U_k \) \((1 ≤ k ≤ n, k \neq i)\) such that \( 0 ≤ |c_k| ≤ ψ \) \((1 ≤ k ≤ n, k \neq i)\). Note that \( m_k, \ell_k \) are not necessarily distinct. Then, \( P_{FT}^{TA} \) is given as
\[ P_{FT}^{TA} = \max_{U_i, W} P_{FT}(U_i, W), \]
where \( W \in W \) and \( U_i \in U \) with \( U_i \not\in W \).

The concept of \((n, ω, ψ, p_1, p_2)\)-group secure and of \((n, ω, ψ, p_1, p_2, p_3, p_4)\)-group full secure signature scheme can now be defined, where \( p_1, p_2, p_3 \) and \( p_4 \) are security parameters whose meanings will be made precise in the following definitions.

**Definition 5.5** Let \( \Pi \) be an Unconditionally Secure Group Signature Scheme opened by TA. Let \( U \) be a set of \( n \) users. Then, \( \Pi \) is \((n, ω, ψ, p_1, p_2)\)-group secure if the following conditions are satisfied: as long as there exist at most \( ω - 1 \) colluders and each user is allowed to generate at most \( ψ \) signatures, the following inequalities hold:
\[ \max\{P_I, P_{G,I}^{TA}, P_S, P_{G,S}^{TA}\} ≤ p_1, \max\{P_T, P_{G,T}^{TA}\} ≤ p_2 \]
where \( P_I \) and \( P_{G,I}^{TA} \) are the probabilities of success in impersonation for user and group signatures respectively, \( P_S \) and \( P_{G,S}^{TA} \) are the probabilities of success in substitution for user and group signatures respectively and \( P_T \) and \( P_{G,T}^{TA} \) are the probabilities of success in transfer with a trap for user and group signatures respectively.

We note that there is an alternative definition of security in which one may use a single security parameter \( p \) instead and define the success probability as
\[ \max\{P_I, P_{G,I}^{TA}, P_S, P_{G,S}^{TA}, P_T, P_{G,T}^{TA}\} ≤ p. \]
In practice, however, some applications may attach more weight to strength against impersonation and substitution than against transfer with a trap, while some other applications may have an emphasis on robustness against transfer with a trap. By introducing two separate parameters $p_1$ and $p_2$, we have an opportunity to design a signature scheme with fine-tuned level of security.

**Definition 5.6** Let $\Pi$ be an Unconditionally Secure Group Signature Scheme opened by TA. Let $\mathcal{U}$ be a set of $n$ users. Then, $\Pi$ is $(n, \omega, \psi, p_1, p_2, p_3, p_4)$-group full secure if it is $(n, \omega, \psi, p_1, p_2)$-group secure and the following inequalities holds:

$$P^{TA}_{FA} \leq p_3, \quad P^{TA}_{FT} \leq p_4$$

where $P^{TA}_{FA}$ and $P^{TA}_{FT}$ are the probabilities of attacks against Full-Anonymity and Full-Traceability properties respectively.

As before, an alternative definition can be used in substitution of Definition 5.6 in which is used a single parameter. Now we prefer to introduce $p_3$ and $p_4$ to give evidence to Full-Anonymity and Full-Traceability properties of a signature scheme.

### 5.2.3 A symmetric construction

In the following, we show an implementation of an Unconditionally Secure Group Signature referred to as symmetric construction. In a symmetric construction signer and verifier use the same secret information both for signing and for verifying ($S = V$). Now we show the simplest construction:

1. **Key Generation and Distribution by TA:** Let $F_{q_0}$ be a finite field with $q_0$ elements such that $q_0 \geq n\omega q$, where $q$ is a security parameter of the system. We assume that the size of $q_0$ is almost the same as $n\omega q$. Then TA divides $F_{q_0}$ into $n$ disjoint subsets $\mathcal{U}_1, \ldots, \mathcal{U}_n$, such that $|\mathcal{U}_\ell| = \omega q$ for $\ell = 1, \ldots, n$. Here the subsets $\mathcal{U}_\ell$ ($\ell = 1, \ldots, n$) are made public for all users. The TA picks uniformly at random a value $u_{TA}$ in $F_{q_0}$, determines a subset $\mathcal{D} = \{d_1, d_2, \ldots, d_{q'}\} \subset F_{q_0}$ with $1 \leq q' \leq n$ and constructs uniformly at random a matrix $A = (a_{i,j,k})$ ($0 \leq i \leq \omega$, $0 \leq j \leq \omega$, $0 \leq k \leq \psi$) such that for any fixed $k$ the matrix $A_k = (a_{i,j})$ is symmetric. Given $A$, the TA defines
the polynomial used for creating the secret information for each user as
\[ F(x, y, z) = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} \sum_{k=0}^{\omega} a_{i,j,k} x^i y^j z^k. \]

For each user \( U_\ell \) (\( 1 \leq \ell \leq n \)) the TA picks uniformly at random a value \( u_\ell \) in \( U_\ell \), an element \( d_\ell \) in \( D \) and calculates
\[ u_{G,\ell} = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,\ell} u_{T_A}^i d_\ell^j + u_\ell \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,\ell} u_{T_A}^i d_\ell^j. \]

The value \( u_{G,\ell} \) represents the group identity of user \( U_\ell \), while \( u_\ell \) represents the user identity. Moreover, we assume that each message \( m \) is an element in \( F_{q_0} \) as well.

For each user \( U_\ell \) (\( 1 \leq \ell \leq n \)), the TA computes his secret information as
\[ e_\ell = \{F(x, u_\ell, z), F(x, u_{G,\ell}, z), u_\ell, u_{G,\ell}\} \]

and sends it to \( U_\ell \) over a secure channel. Once the secret information has been delivered, there is now no need for the TA to keep it. However, he must store \( u_{T_A} \) and must define a way to retrieve the value \( t \) from \( u_{G,\ell} \) (for example he can use a table with an entry for each pair \((u_{G,\ell}, t)\)).

2. **Signature Generation**: For a message \( m \in F_{q_0} \) the user \( U_s \) can generate both a user signature or a group signature. There are two cases:

- \( U_s \) wants to generate a user signature.
  Then he computes \( \alpha = \{a_{s,m}(x), u_s\} \) using his secret information, where \( a_{s,m}(x) = F(x, u_s, m) \) and sends \((m, \alpha)\) with his identity \( U_s \).

- \( U_s \) wants to generate a group signature.
  Then he computes \( \alpha = \{a_{s,m}(x), u_{G,s}\} \) using his secret information, where \( a_{s,m}(x) = F(x, u_{G,s}, m) \) and sends \((m, \alpha)\) without including his identity.

3. **Signature Verification**: On receiving the signature \((m, \alpha)\), the user \( U_r \) checks if it is a user or a group signature. The difference between user and group signature is in the presence of the identity of the sender (in our case \( U_s \)). There are two cases:

- User Signature:
Chapter 5. Unconditionally Secure Group Signatures

If \((m, \alpha)\) is a signature of \(U_s\), then user \(U_r\) checks whether \(\alpha\) is valid or not, by using his secret information \(e_r\). Specifically, \(U_r\) accepts \((m, \alpha)\) as being a valid message-signature pair from \(U_s\) if

\[ F(x, u_r, z) | x = u_s, z = m = a_{s,m}(x) | x = u_r \]

and \(u_s \in U_s\).

- **Group Signature**

If \((m, \alpha)\) is a group signature, then user \(U_r\) checks the validity of the signature by using the group identity of the signer and his secret information \(e_r\). Specifically, \(U_r\) accepts \((m, \alpha)\) as being a valid message-group signature if

\[ F(x, u_r, z) | x = u_{G,s}, z = m = a_{s,m}(x) | x = u_r. \]

4. **Open a signature:** In special situations, a member of the group that has received a group signature might need to know the real identity of the signer. Signer’s identity can be computed by the TA from user’s group identity. More precisely, if user \(U_r\) wants to know the signer’s identity of a received group signature, he sends the value \(u_{G,s}\) to the TA. The TA retrieves the value \(d_t \in D\) from \(u_{G,s}\) and since

\[ u_{G,s} = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,t-1} u_{TA}^{i} d_t^i + u_s \sum_{i=0}^{\omega} a_{s,j,t} u_{TA}^i d_t^i \]

he computes:

\[ u_s = \frac{u_{G,s} - \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,t-1} u_{TA}^{i} d_t^i}{\sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,t} u_{TA}^i d_t^i}. \]

**Theorem 5.1** The required memory size in the above construction is given as follows:

\[ |S| = |V| = 2^\omega q q_0^{(\omega+1)(\psi+1)+1} \]

\[ |A| = \begin{cases} \omega q q_0^{\omega+1} & \text{for a user signature} \\ q_0^{\omega+2} & \text{for a group signature.} \end{cases} \]
5.2.4 Reducing memory storage of the symmetric construction

The solution that we have proposed for the symmetric scheme is simple, but the size of the secret information is too large. In the following we propose a better construction.

1. Key Generation and Distribution by TA: This phase is like that of the previous construction. Let $F_{q_0}$ be a finite field with $q_0$ elements such that $q_0 \geq n\omega q$, where $q$ is a security parameter of the system. We assume that the size of $q_0$ is almost the same as $n\omega q$. Then TA divides $F_{q_0}$ into $n$ disjoint subsets $U_1, \ldots, U_n$, such that $|U_\ell| = \omega q$ for $\ell = 1, \ldots, n$. Here the subsets $U_\ell$ ($\ell = 1, \ldots, n$) are made public for all users. The TA picks uniformly at random a value $u_{TA}$ in $F_{q_0}$, determines a subset $D = \{d_1, d_2, \ldots, d_{q'}\} \subset F_{q_0}$ with $1 \leq q' \leq n$ and constructs uniformly at random a matrix $A = (a_{i,j,k})$ ($0 \leq i \leq \omega$, $0 \leq j \leq \omega$, $0 \leq k \leq \psi$) such that for any $k$ the matrix $A_k = (a_{i,j})$ is symmetric. Given $A$, the TA can define the polynomial used for creating the secret information for each user as

$$F(x,y,z) = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} \sum_{k=0}^{\psi} a_{i,j,k} x^i y^j z^k.$$

For each user $U_\ell$ ($1 \leq \ell \leq n$) the TA picks uniformly at random a value $u_\ell$ in $F_{q_0}$, an element $d_\ell$ in $D$ and calculates

$$u_{G,\ell} = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,\ell-1} u_{TA}^i d_\ell^j + u_\ell \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,\ell} u_{TA}^i d_\ell^j.$$

The value $u_{G,\ell}$ represents the group identity of user $U_\ell$, while $u_\ell$ represents the user identity. Moreover, we assume that each message $m$ is an element in $F_{q_0}$ as well. For each user $U_\ell$ ($1 \leq \ell \leq n$), the TA computes his secret information as

$$e_\ell = \{F(x,u_\ell u_{G,\ell},z), u_\ell, u_{G,\ell}\}$$

and sends it to $U_\ell$ over a secure channel. Once the secret information has been delivered, there is now no need for the TA to keep it. However, he must store $u_{TA}$ and must define a way to retrieve the value $t$ from $u_{G,\ell}$ (for example he can use a table with an entry for each pair $(u_{G,\ell}, t)$).

2. Signature Generation: For a message $m \in F_{q_0}$, the user $U_s$ can choose to send a user signature or a group signature.
• If $U_s$ wants to generate a user signature, then he computes $F(x, u_s, z)$ using his secret information in the following way

$$F(x, u_s, z) = F(x, u_s u_{G,s}, z) \oplus (1, 1/u_{G,s}, 1).$$

The signature of message $m$ is $\alpha = \{a_{s,m}(x), u_s\}$ where $a_{s,m}(x) = F(x, u_s, m)$. Then, $U_s$ sent the pair $(m, \alpha)$ including his identity.

• Otherwise, if $U_s$ wants to generate a group signature, then he computes $F(x, u_{G,s}, z)$ using his secret information in the following way

$$F(x, u_{G,s}, z) = F(x, u_s u_{G,s}, z) \oplus (1, 1/u_s, 1).$$

The signature of message $m$ is $\alpha = \{a_{s,m}(x), u_{G,s}\}$ where $a_{s,m}(x) = F(x, u_{G,s}, m)$. Then, $U_s$ sent the pair $(m, \alpha)$ without including his identity.

3. **Signature Verification**: On receiving a pair $(m, \alpha)$, the user $U_r$ checks if it is a user or a group signature. This can be seen by checking the presence of the signer’s identity. By using his secret information $e_r$, $U_r$ checks whether $\alpha$ is valid or not. More precisely, let $\bar{u}_s$ defined as:

$$\bar{u}_s = \begin{cases} u_s & \text{if } (m, \alpha) \text{ is a user signature} \\ u_{G,s} & \text{if } (m, \alpha) \text{ is a group signature} \end{cases}$$

then $U_r$ accepts $(m, \alpha)$ as being a valid message-signature pair if

$$F(x, u_r u_{G,r}, z)|_{x=\bar{u}_s, z=m} = a_{s,m}(x)|_{x=u_r u_{G,r}}$$

and in the case of a user signature, $U_r$ checks also that $u_s \in U_s$.

4. **Open a signature**: Once $U_r$ has received a valid group signature, he can ask the TA to open the signature and reveals the real signer’s identity. To do this, $U_r$ must send to TA the group identity $u_{G,s}$ of the signature. Given the value $u_{G,s}$, the TA retrieves the value $d_t$ and since

$$u_{G,s} = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,t-1} u_{TA}^i d_t^j + u_s \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,t} u_{TA}^i d_t^j,$$

he computes:

$$u_s = \frac{u_{G,s} - \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,t-1} u_{TA}^i d_t^j}{\sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,t} u_{TA}^i d_t^j}.$$
Chapter 5. Unconditionally Secure Group Signatures

**Theorem 5.2** The required memory size in the above construction is given as follows:

\[ |S| = |V| = \omega q q_0^{(\omega+1)(\psi+1)+1} \]

\[ |A| = \begin{cases} \omega q q_0^{\psi+1} & \text{for a user signature} \\ q_0^{\omega+2} & \text{for a group signature.} \end{cases} \]

**Theorem 5.3** The above symmetric signature scheme results in a \( (n, \omega, \psi, \frac{1}{q_0}, \frac{1}{q}, \frac{2(\omega+1)}{q_0 q'}) \)-group full secure scheme.

**Proof.** To prove this, we have to compute the probabilities of the symmetric scheme.

- **Impersonation attack for user signature:** Assume that after seeing a signed message \((m_{i_0}, \alpha)\) published by \(U_{i_0}\), the colluders \(U_1, \ldots, U_{\omega-1}\) want to generate \((m_{i_1}, \alpha')\), such that \(m_{i_1} = m_{i_0}\) and the user \(U_{i_2}\) will accept it as a valid signed message of the user \(U_{i_1}\), i.e. \(\alpha'\) consists of \(\{u_{i_1}', a_{i_1, m_{i_1}}(x)\}\) such that \(a_{i_1, m_{i_1}}(u_{i_2}) = F(u_{i_1}', u_{i_2} u_{G, i_2}, m)\) and \(u_{i_1}' \in U_{i_1}\). Remember that in the phase *Key Generation and Distribution by TA* of the scheme we have indicate with \(A_k\) the symmetric matrix obtained by \(A\) for a fixed \(k\).

Letting

\[ P_k(x, y) = \begin{pmatrix} 1 \\ y \\ y^2 \\ \vdots \\ y^\omega \end{pmatrix} \begin{pmatrix} 1 \\ y \\ y^2 \\ \vdots \\ y^\omega \end{pmatrix} \]

for \(k = 0, \ldots, \psi\), we can represent the polynomial \(F(x, y, z)\) as

\[ F(x, y, z) = \sum_{k=0}^{\psi} P_k(x, y) z^k. \]

The knowledge of the coalition can be represented by a \((\omega + 1) \times \omega\) matrix \(D\), where

\[ D = \left( \sum_{k=0}^{\psi} A_k m_{i_0}^k \right) \begin{pmatrix} 1 & 1 & \cdots & 1 \\ U_{i_0} & U_1 & \cdots & U_{\omega-1} \\ U_{i_0}^2 & U_1^2 & \cdots & U_{\omega-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ U_{i_0}^\omega & U_1^\omega & \cdots & U_{\omega-1}^\omega \end{pmatrix}. \]
From Lemma 3.1, there exist $q_0$ different matrices $X$ such that

$$D = X \begin{pmatrix} 1 & 1 & \cdots & 1 \\ U_{i_0} & U_1 & \cdots & U_{\omega-1} \\ \vdots & \vdots & \ddots & \vdots \\ U_{\omega-1} & U_{\omega} & \cdots & U_{\omega} \end{pmatrix}.$$ 

This implies that there are $q_0$ different values for $\sum_{k=0}^{\psi} A_k m_{i_0}^k$.

In order for the colluders to succeed the attack, they need to find a pair of $u'_{i_1}$ and $a'_{i_1,m_{i_1}}(x)$ such that

$$a'_{i_1,m_{i_1}}(u_{i_2}) = \left( 1, u'_{i_1}, u'_{i_1}, \ldots, u'_{i_1} \right) \left( \sum_{k=0}^{\psi} A_k m_{i_0}^k \right) \begin{pmatrix} 1 \\ u_{i_2} \\ u_{i_2}^2 \\ \vdots \\ u_{i_2}^\omega \end{pmatrix}$$

and $u'_{i_1} \in U_{i_1}$. Letting $d$ be

$$d = \left( 1, u'_{i_1}, u'_{i_1}, \ldots, u'_{i_1} \right) \left( \sum_{k=0}^{\psi} A_k m_{i_0}^k \right) \begin{pmatrix} 1 \\ u_{i_2} \\ u_{i_2}^2 \\ \vdots \\ u_{i_2}^\omega \end{pmatrix}$$

$q_0$ different matrices for $\sum_{k=0}^{\psi} A_k m_{i_0}^k$ result in $q_0$ different values for $d$. This indicates that the probability of succeeding in finding $a'_{i_1,m_{i_1}}(x)$, such that $a'_{i_1,m_{i_1}}(u_{i_2}) = d$, is $\frac{1}{q_0}$, i.e. $P_I = \frac{1}{q_0}$.

**Impersonation attack for group signature:** Assume that after seeing a user signed message $(m_{i_0}, \alpha)$ published by $U_{i_0}$, the colluders $U_1, \ldots, U_{\omega-1}$ want to generate $(m_{i_1}, \alpha')$, such that $m_{i_1} = m_{i_0}$ and the user $U_{i_2}$ and the TA will accept it as a valid group signed message of the user $U_{i_1}$, i.e. $\alpha'$ consists of $\{u'_{i_1}, a'_{i_1,m_{i_1}}(x)\}$ such that $a'_{i_1,m_{i_1}}(x) = F(x, u'_{G,i_1}, m_{i_1})$ and

$$u'_{G,i_1} = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,t-1} u'_{T,A} d'_i + u_i \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,t} u'_{T,A} d'_i.$$
To realize an Impersonation attack for group signature, the coalition must find the group identity of $U_{i_1}$ and a valid group signature generated with this group identity. Remember that a group identity is a signature produced by TA using the scheme of Section 4.2.2 and evaluated in an element of $\mathcal{D}$ in which the signed message is the user’ identity $u_{i_1}$. For the sake of simplicity, denote with $P_0(x, y) = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,t} x^i y^j$ and with $P_1(x, y) = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,t} x^i y^j$, then we can represent $u'_{G,i_1}$ as:

$$u'_{G,i_1} = P_0(u_{TA}, d_t) + u_{i_1} P_1(u_{TA}, d_t).$$

Since each identity of $U_{i_1}$ is an element of $U_{i_1}$ and since $|U_{i_1}| = \omega q$, then the probability that $u_{i_1}$ is a user’s identity of $U_{i_1}$ is

$$Pr(u_{i_1} \in U_{i_1}) = \frac{1}{\omega q}.\$$

Given $u_{i_1}$, the colluders must generate a signature produced by TA of message $u_{i_1}$ to obtain a group identity of $U_{i_1}$. From Theorem 4.1 (Substitution attack case), we know that the success probability of finding a signature produced by TA on message $u_{i_1}$ is less equal to $\frac{1}{q_0}$ and since the colluders can find the value $d_t \in \mathcal{D}$ (that was chosen randomly by the TA) with probability $\frac{1}{q}$, we can conclude that the probability of finding a valid group identity of $U_{i_1}$ is

$$Pr(u'_{G,i_1} \text{ is a valid group identity of } U_{i_1}) \leq \frac{1}{q' \omega q q_0}.\$$

The value $u_{G,i_1}$ can be considered as a starting point to realize an impersonation attack as in Theorem 4.1 and, hence,

$$P^{TA}_{G,i} \leq \frac{1}{q' \omega q q_0}.\$$

- **Substitution attack for user signature:** The proof for the substitution attack for user signature is similar to that presented in the case Impersonation attack for user signature. So similarly, we can prove that $P_S \leq \frac{1}{q_0}$.

- **Substitution attack for group signature:** The proof for the substitution attack for group signature is similar to that of Impersonation attack for group signature. So similarly, we can prove that:

$$P^{TA}_{G,S} \leq \frac{1}{q' \omega q q_0}.$$
Chapter 5. Unconditionally Secure Group Signatures

- **Transfer with a trap attack for user signature:** Assume that after seeing a signed message \((m_{i_0}, \alpha)\) published by \(U_{i_0}\), the colluders \(U_1, \ldots, U_{\omega-1}\) want to generate \((m_{i_0}, \alpha')\), such that \(\alpha' \neq \alpha\) and the user \(U_{i_1}\) will accept it as a valid signed message of the user \(U_{i_0}\). Let \(\alpha = \{u_{i_0}, a_{i_0,m_{i_0}}(x)\}\) as described before. Since \(a_{i_0,m_{i_0}}(x)\) is a polynomial in the variable \(x\) of degree at most \(\omega\), then \(a_{i_0,m_{i_0}}(x)\) \((a_{i_0,m_{i_0}}(x) \neq a_{i_0,m_{i_0}}(x))\) has at most \(\omega\) pairs \(\{c, a_{i_0,m_{i_0}}(c)\}\), for which \(c \in F_{q_0}\) and \(a_{i_0,m_{i_0}}(c) = a_{i_0,m_{i_0}}(c)\), where \(a_{i_0,m_{i_0}}(x)\) is a polynomial in \(x\) of degree at most \(\omega\). Hence, the best strategy for succeeding transfer with a trap is as follows: The colluders choose uniformly at random \(\omega\) distinct numbers \(u_{i_1}, \ldots, u_{i_1}^{(\omega)}\) from \(U_{i_1}\) and generate \(a'_{i_0,m_{i_0}}(x) = a_{i_0,m_{i_0}}(x)\) \((a'_{i_0,m_{i_0}}(x) \neq a_{i_0,m_{i_0}}(x))\) such that \(a'_{i_0,m_{i_0}}(u_{i_1}) = a_{i_0,m_{i_0}}(u_{i_1}), a'_{i_0,m_{i_0}}(u_{i_1}^{(2)}) = a_{i_0,m_{i_0}}(u_{i_1}^{(2)}), \ldots, a'_{i_0,m_{i_0}}(u_{i_1}^{(\omega)}) = a_{i_0,m_{i_0}}(u_{i_1}^{(\omega)}).\) Then, the colluders send \(\alpha' = \{u_{i_0}, a'_{i_0,m_{i_0}}(x)\}\) to \(U_{i_1}\). The attack is successful if and only if \(u_{i_1} \in \{u_{i_1}^{(1)}, u_{i_1}^{(2)}, \ldots, u_{i_1}^{(\omega)}\}\). Hence, \(P_T = \frac{\omega}{q_0} = \frac{1}{q}\).

- **Transfer with a trap attack for group signature:** Assume that after seeing a user signed message \((m_{i_0}, \alpha)\) published by \(U_{i_0}\), the colluders \(U_1, \ldots, U_{\omega-1}\) want to generate a group signature \((m_{i_0}, \alpha')\), such that \(\alpha' \neq \alpha\) and the user \(U_{i_1}\) and the TA will accept it as a valid group signed message of the user \(U_{i_0}\). Let \(\alpha = \{u_{i_0}, a_{i_0,m_{i_0}}(x)\}\) where \(a_{i_0,m_{i_0}} = F(x, u_{i_0}, m_{i_0})\). The first step that colluders can do is to find a valid group identity of user \(U_{i_0}\). To do this, they can compute a signature produced by TA on user’s identity \(u_{i_0}\) using the scheme presented in Section 4.2.2 and evaluate it at \(d_t \in D\). Remember that we indicate with \(u_{G,i_0}\) \(U_{i_0}\)’s group identity and that we have defined in the case Impersonation attack for group signature \(P_0(x, y) = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,t-1} x^i y^j\) and \(P_1(x, y) = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,t} x^i y^j\). To compute a TA’s signature on user’s identity \(u_{i_0}\) the colluders must realize a substitution attack on the scheme presented in Section 4.2.2. Based on Theorem 4.1 we know that the success probability of a substitution attack is less equal to \(\frac{1}{q_0}\) and since the value \(d_t\) is chosen randomly by the TA, the probability of finding a value \(d_t' \in D\) such that \(P_0(u_{TA}, d_t') + u_{i_0} P_1(u_{TA}, d_t') = u_{G,i_0}\) is \(\frac{1}{q}\). Then we can say that the probability to find a valid group identity \(u_{G,i_0}\) of \(U_{i_0}\) is

\[
Pr(u_{G,i_0} \text{ is a valid group identity of } U_{i_0}) \leq \frac{1}{q q_0}.
\]

Given \(u_{G,i_0}\), the colluders must compute the signature \((m_{i_0}, \alpha'')\) such that \(\alpha'' = \ldots\)
(u_{G,i_0}, a_{u_{G,i_0},m_{i_0}}(x)) and \(a_{u_{G,i_0},m_{i_0}}(x) = F(x, u_{G,i_0}, m_{i_0})\). They can obtain the value \(\alpha''\) realizing an impersonation attack against a user whose identity is \(u_{G,i_0}\) with success probability is \(\frac{1}{q_0}\). Now, the pair \((m_{i_0}, \alpha'')\) can be consider as a starting point for a transfer with a trap attack for user signature whose success probability is \(\frac{1}{q}\).

Then the \(P_{G,T}\) of the scheme is:

\[
P_{G,T}^{TA} \leq \frac{1}{q_0 q}.
\]

- **Full-Anonymous:** To prove the Full-Anonymous property of the scheme, we calculate the probability of finding the identity of the signer from the group identity. Recall from Definition 5.4 that \(P_{FT}^{TA}(W,U_i)\) is defined as

\[
P_{FT}^{TA}(W,U_i) = \max_{s_W, v_W} \max_{1 \leq k \leq n} \max_{c_k = \{(m_{k_1}, \alpha_{k_1})\}} \max_{\text{GId}=\{u_{i, \text{Gid}}\}} \text{Pr}(W \text{ identifies } U_i | u_{G,i}, s_W, v_W, \{c_k\}, \text{GId}),
\]

where \(c_k\) is a set of user signed messages and \(\text{GId}\) is a set of group identities known by the coalition. Since in the proposed construction group identities are not used in user signatures and since they are linearly independent we have that

\[
P_{FT}^{TA}(W,U_i) = \text{Pr}(W \text{ identifies } U_i | u_{G,i}).
\]

As we have seen in Section 5.2.4, the group identity of user \(U_\ell\) is obtained by TA in the following way:

\[
u_{G,\ell} = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,\ell-1} u_{TA}^i d_\ell^j + u_\ell \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,\ell} u_{TA}^i d_\ell^j
\]

where \(u_{TA}\) is the identity of TA, \(d_\ell\) is an element of \(D\) chosen at random by TA and \((a_{ijk})\) are the coefficients of the polynomial \(F(x,y,z)\) used to calculate the secret information of each user. Let indicate with \(F_h(u_{TA}, y) = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,h} u_{TA}^i y^j\) a polynomial of degree \(\omega\). Then we can represent the group identity of \(U_\ell\) as

\[
u_{G,\ell} = F_{\ell-1}(u_{TA}, d_\ell) + u_\ell F_{\ell}(u_{TA}, d_\ell).
\]

Since each polynomial \(F_i(u_{TA}, y)\) for \(i = t-1, t\) has degree \(\omega\), then each polynomial has \(\omega + 1\) coefficients, so the probability of finding \(F_{\ell-1}(u_{TA}, y)\) and \(F_{\ell}(u_{TA}, y)\) is

\[
\text{Pr}(F_{\ell-1}(u_{TA}, y), F_{\ell}(u_{TA}, y) | u_{G,\ell}) \leq \frac{2(\omega + 1)}{q_0}.
\]
As we have seen in the Key Generation and Distribution by TA of Section 5.2.4, the TA chooses at random an element of $D$ to calculate the group identity of a member. Since $|D| = q'$, the probability of finding the value $d_t$ used for user $U_\ell$ is

$$Pr(d_t|u_{G,\ell}) = \frac{1}{q'}.$$ 

Hence, the probability of recovering the user identity from the group identity is

$$P_{TA}^{FA} = Pr(u_\ell|u_{G,\ell}) \leq \frac{2(\omega + 1)}{q_0 q'}.$$ 

- **Full-Traceability:** To prove the Full-Traceability property we calculate the probability to create signature that cannot be traced back to any member of the coalition. Let consider the case in which a user $U_i$ generate a group signature accepted by $U_j$ but such that in the Open a signature phase of Section 5.2.4 the TA cannot return a valid user identity. Let consider a group identity $u_{G,\ell}$ chose at random in $F_{q_0}$ not used by any member of the coalition. The probability of finding such a group identity is

$$Pr(u_{G,\ell}| u_{G,\ell} \text{ is not used in } U) \leq 1 - \frac{n}{q_0}.$$ 

The coalition $W$ can do an impersonation attack against user $U_j$ such that $U_j$ accepts the signature. From the Impersonation attack for user signature case, we know that the probability of such an event is:

$$P_I = \frac{1}{q_0}.$$ 

Since the group identity $u_{G,\ell}$ is not associated to any group member, then in the Open a signature phase the TA cannot return any valid user identity. Hence,

$$P_{FT}^{TA} \leq \left(1 - \frac{n}{q_0}\right) \frac{1}{q_0}.$$ 

Using the Definition 5.6 we have the proof.

### 5.2.5 An asymmetric construction

In this Section, we propose an asymmetric construction of an Unconditionally Secure Group Signature scheme in which the secret information for signing and that for verifying are different. As before, we propose initially a simple solution and then an optimized one.
1. **Key Pair Generation and Distribution by TA:** Let $F_q$ be a finite field with $q$ elements such that $q \geq n$. The TA picks $n + 1$ elements $v_{TA}, v_1, v_2, \ldots, v_n$ uniformly at random in $F_q^{\omega-1}$ for himself and for users $U_1, U_2, \ldots, U_n$ respectively, determines a subset $\mathcal{D} = \{d_1, d_2, \ldots, d_q\} \subset F_q$ with $1 \leq q' \leq n$ and constructs a polynomial $G(x, y_1, \ldots, y_{\omega-1}, z)$ as follows:

$$G(x, y_1, \ldots, y_{\omega-1}, z) = \sum_{i=0}^\omega \psi \sum_{k=0}^i a_{ijk} x^i y_j^i z^k$$

where the coefficients $a_{ijk}$ ($0 \leq i \leq \omega$, $1 \leq j \leq \omega - 1$ and $0 \leq k \leq \psi$) are chosen uniformly at random from $F_q$. Moreover, we assume that each user’s identity $U_\ell$ and message $m$ are elements of $F_q$. For each user $U_\ell$ ($1 \leq \ell \leq n$), the TA chooses uniformly at random an element $d_t$ in $\mathcal{D}$ used to compute the group identity $U_{G,\ell} = \sum_{i=0}^\omega d_i + U_\ell \sum_{i=0}^\omega \sum_{j=1}^{\omega-1} a_{ijt} d_j v_{TA,j}$, computes the signing key for user signature $s_\ell = G(U_\ell, y_1, \ldots, y_{\omega-1}, z)$, the signing key for group signature $\hat{s}_\ell = G(U_{G,\ell}, y_1, \ldots, y_{\omega-1}, z)$ and the verification key $\hat{v}_\ell = G(x, v_\ell, z)$. Then $U_\ell$’s secret information is

$$e_\ell = \{s_\ell, \hat{s}_\ell, v_\ell, \hat{v}_\ell, U_{G,\ell}\}$$

that the TA sends to $U_\ell$ over a secure channel. Once all the keys are delivered, there is no need for the TA to keep the users’ secret information. He must take only $v_{TA}$ and must define a way to retrieve the value $t$ from $U_{G,\ell}$ (for example he can use a table with an entry for each pair $(U_{G,\ell}, t)$).

2. **Signature Generation:** For a message $m \in F_q$ the user $U_{\ell_1}$ can generate both a user signature or a group signature. There are two cases:

- if $U_{\ell_1}$ wants to generate a user signature then using his signing key $s_{\ell_1}$, he calculates

$$\alpha = s_{\ell_1}|_{z=m} = G(U_{\ell_1}, y_1, \ldots, y_{\omega-1}, z)|_{z=m} = G(U_{\ell_1}, y_1, \ldots, y_{\omega-1}, m).$$

Then the tuple $(m, \alpha, U_{\ell_1})$ is sent by $U_{\ell_1}$.

- otherwise, if $U_{\ell_1}$ wants to generate a group signature then using his signing key $\hat{s}_{\ell_1}$, he calculates

$$\alpha = \hat{s}_{\ell_1}|_{z=m} = G(U_{G,\ell_1}, y_1, \ldots, y_{\omega-1}, z)|_{z=m} = G(U_{G,\ell_1}, y_1, \ldots, y_{\omega-1}, m).$$
Then the tuple \((m, \alpha, U_{G, \ell_1})\) is sent by \(U_{\ell_1}\).

3. **Signature Verification**: On receiving the pair \((m, \alpha)\), the user \(U_{\ell_2}\) must check if it is a user or a group signature. As before, this can be seen by the presence of signer’s identity. By using the pair \((v_{\ell_2}, \tilde{v}_{\ell_2})\) included in his secret information, user \(U_{\ell_2}\) checks whether \(\alpha\) is valid or not. Specifically, let \(\bar{U}_{\ell_1}\) defined as:

\[
\bar{U}_{\ell_1} = \begin{cases} 
U_{\ell_1} & \text{if } (m, \alpha) \text{ is a user signature} \\
U_{G, \ell_1} & \text{if } (m, \alpha) \text{ is a group signature}
\end{cases}
\]

then \(U_{\ell_2}\) accept \((m, \alpha)\) as being a valid message-signature pair if

\[
\tilde{v}_{\ell_2}|_{x = c_{\ell_1}, z = m} = G(x, v_{\ell_2}, z)|_{x = c_{\ell_1}, z = m} = \alpha|_{(y_1, y_2, \ldots, y_{\omega-1}) = (v_{1, \ell_2}, \ldots, v_{\omega-1, \ell_2})}.
\]

4. **Open a Signature**: Signer’s identity can be computed by the TA from user’s group identity. More precisely, if user \(U_{\ell_2}\) wants to know signer’s identity of a group signature, he must send the value \(U_{G, \ell_1}\) to the TA. Given the value \(U_{G, \ell_1}\), the TA retrieves the value \(d_t\) and since \(U_{G, \ell_1} = \sum_{i=0}^{\omega} d_t^i + U_{\ell_1} \sum_{j=0}^{\omega-1} \sum_{i=1}^{m} a_{ij} d_t^i v_{TA,j}\), he computes signer’s identity as:

\[
U_{\ell_1} = \frac{U_{G, \ell_1} - \sum_{i=0}^{\omega} d_t^i}{\sum_{i=0}^{\omega} \sum_{j=1}^{m-1} a_{ij} d_t^i v_{TA,j}}.
\]

**Theorem 5.4** The required memory size in the above construction is given as follows:

\[
|S| = 2q^{2(\psi+1)+1}
\]

\[
|V| = q^{2(\omega+1)(\psi+2)-2}
\]

\[
|A| = q^{\omega+1}.
\]

5.2.6 Reducing memory storage of the asymmetric construction

As for the symmetric solution, we propose another solution to the problem in which the secret information that every user has to store is smaller in size than in the simple solution presented in the previous Section.
Chapter 5. Unconditionally Secure Group Signatures

1. Key Pair Generation and Distribution by TA: This step is similar to that of the previous scheme. Let $F_q$ be a finite field with $q$ elements such that $q \geq n$. The TA picks $n+1$ elements $v_{TA}, v_1, v_2, \ldots, v_n$ uniformly at random in $F_{q-1}$ for himself and for users $U_1, U_2, \ldots, U_n$ respectively, determines a subset $D = \{d_1, d_2, \ldots, d_q'\} \subset F_q$ with $1 \leq q' \leq n$ and constructs a polynomial $G(x, y_1, \ldots, y_{\omega-1}, z)$ as follows:

$$G(x, y_1, \ldots, y_{\omega-1}, z) = \sum_{i=0}^{\omega} \sum_{k=0}^{\psi} a_{0k} x^i z^k + \sum_{i=0}^{\omega} \sum_{j=1}^{\omega-1} \sum_{k=0}^{\psi} a_{ijk} x^i y_j z^k$$

where the coefficients $a_{ijk}$ ($0 \leq i \leq \omega$, $1 \leq j \leq \omega - 1$ and $0 \leq k \leq \psi$) are chosen uniformly at random from $F_q$. Moreover, we assume that each user’s identity $U_\ell$ and message $m$ are elements of $F_q$. For each user $U_\ell$ ($1 \leq \ell \leq n$), the TA chooses uniformly at random an element $d_\ell$ in $D$ used to compute the group identity $U_{G,\ell} = \sum_{i=0}^{\omega} d_\ell^i + U_\ell \sum_{i=0}^{\omega} \sum_{j=1}^{\omega-1} a_{ij} d_\ell^i v_{TA,j}$, computes the signing key $s_\ell = G(U_\ell U_{G,\ell}, y_1, \ldots, y_{\omega-1}, z)$ and the verification key $\tilde{\nu}_\ell = G(x, v_{\ell}, z)$. Then $U_\ell$’s secret information is

$$e_\ell = \{s_\ell, v_{\ell}, \tilde{\nu}_\ell, U_{G,\ell}\}$$

that the TA sends to $U_\ell$ over a secure channel. Once all the keys are delivered, there is no need for the TA to keep the users’ secret information. He must take only $v_{TA}$ and must define a way to retrieve the value $t$ from $U_{G,\ell}$ (for example he can use a table with an entry for each pair $(U_{G,\ell}, t)$).

2. Signature Generation: For a message $m \in F_q$ the user $U_{\ell_1}$ can generate both a user signature or a group signature. There are two cases:

- if $U_{\ell_1}$ wants to generate a user signature then using his signing key $s_{\ell_1}$, he computes $G(U_{\ell_1}, y_1, \ldots, y_{\omega-1}, z)$ as:

$$G(U_{\ell_1}, y_1, \ldots, y_{\omega-1}, z) = G(U_{\ell_1} U_{G,\ell_1}, y_1, \ldots, y_{\omega-1}, z) \oplus (1/U_{G,\ell_1}, 1, \ldots, 1)$$

and

$$\alpha = G(U_{\ell_1}, y_1, \ldots, y_{\omega-1}, z)|_{z=m} = G(U_{\ell_1}, y_1, \ldots, y_{\omega-1}, m).$$

Then the tuple $(m, \alpha, U_{\ell_1})$ is sent by $U_{\ell_1}$. 
Chapter 5. Unconditionally Secure Group Signatures

• otherwise, if $U_{\ell_1}$ wants to generate a group signature then using his signing key $s_{\ell_1}$, he computes $G(U_{G,\ell_1}, y_1, \ldots, y_{\omega-1}, z)$ as:

$$G(U_{G,\ell_1}, y_1, \ldots, y_{\omega-1}, z) = G(U_{\ell_1}U_{G,\ell_1}, y_1, \ldots, y_{\omega-1}, z) \oplus (1/U_{\ell_1}, 1, \ldots, 1, 1)$$

and

$$\alpha = G(U_{G,\ell_1}, y_1, \ldots, y_{\omega-1}, z)|_{z=m} = G(U_{G,\ell_1}, y_1, \ldots, y_{\omega-1}, m).$$

Then the tuple $(m, \alpha, U_{G,\ell_1})$ is sent by $U_{\ell_1}$.

3. Signature Verification: On receiving the pair $(m, \alpha)$, the user $U_{\ell_2}$ must check if it is a user or a group signature. As before, this can be seen by the presence of the signer’s identity. By using his verification key $\tilde{v}_{\ell_2}$, $U_{\ell_2}$ checks whether $\alpha$ is valid or not. Specifically, let

$$\tilde{U}_{\ell_1} = \begin{cases} U_{\ell_1} & \text{if } (m, \alpha) \text{ is a user signature} \\ U_{G,\ell_1} & \text{if } (m, \alpha) \text{ is a group signature.} \end{cases}$$

$U_{\ell_2}$ accept $(m, \alpha)$ as being a valid message-signature pair if

$$\tilde{v}_{\ell_2}|_{x=\tilde{U}_{\ell_1}, z=m} = G(x, v_{\ell_2}, z)|_{x=\tilde{U}_{\ell_1}, z=m} = \alpha|_{(y_1,y_2,\ldots,y_{\omega-1})=(v_{\ell_2},\ldots,v_{\omega-1})}.$$ 

4. Open a signature: Signer’s identity can be computed by the TA from user’s group identity. More precisely, if user $U_{\ell_2}$ wants to know signer’s identity of a group signature, he must send the value $U_{G,\ell_1}$ to the TA. Given the value $U_{G,\ell_1}$, the TA retrieves the value $d_t$ and since $U_{G,\ell_1} = \sum_{i=0}^{\omega} d_i + U_{\ell_1} \sum_{i=0}^{\omega} \sum_{j=1}^{\omega-1} a_{ij} d_i v_{TA,j}$, he computes signer’s identity as:

$$U_{\ell_1} = \frac{U_{G,\ell_1} - \sum_{i=0}^{\omega} d_i}{\sum_{i=0}^{\omega} \sum_{j=1}^{\omega-1} a_{ij} d_i v_{TA,j}}.$$ 

**Theorem 5.5** The required memory size in the above construction is given as follows:

$$|S| = q^{\omega(\psi+1)+1}$$

$$|V| = q^{2(\omega+1)(\psi+2)-2}$$

$$|A| = q^{\omega+1}.$$
Theorem 5.6  The above asymmetric scheme for group signatures results in a \((n, \omega, \psi, \frac{2}{q} - \frac{1}{q^{2}}, \frac{1}{q^{2}}, \frac{\omega + 1}{q^{2}}, \frac{2}{q^{2}} - \frac{1}{q^{2}})\)-group full secure scheme.

**Proof.** To prove the result, we need to compute the \(P_{I}, P_{G,I}, P_{S}, P_{G,S}, P_{T}, P_{G,T}, P_{F,A}\) and \(P_{F,T}\) probabilities of the asymmetric scheme. Along the same lines of the proof of Theorem 5.3, we can prove that \(P_{I} = \frac{2}{q} - \frac{1}{q^{2}}, P_{S} \leq \frac{2}{q} - \frac{1}{q^{2}}\) and \(P_{T} = \frac{1}{q}\). In the following we give only the proof for \(P_{G,I}, P_{G,S}\) and \(P_{G,T}\).

- **Impersonation attack for group signature:** Assume that after seeing a user signed message \((m_{i0}, \alpha)\) published by \(U_{i0}\), the colluders \(U_{1}, \ldots, U_{\omega-1}\) want to generate \((m_{i1}, \alpha')\), such that \(m_{i1} = m_{i0}\) and the user \(U_{i2}\) and the TA will accept it as a valid group signed message of the user \(U_{i1}\), i.e. \(\alpha' = G(U_{G,i1}, y_{1}, \ldots, y_{\omega-1}, m_{i1})\) where \(U_{G,i1} = \sum_{i=0}^{\omega} d_{i} + U_{i1} \sum_{i=0}^{\omega-1} \sum_{j=1}^{\omega} a_{ij} d_{i} v_{TA,j}\).

To realize an **Impersonation attack for group signature**, the colluders must find the group identity of \(U_{i1}\) and a valid group signature generated with this group identity. Remember that a group identity is a signature produced by TA using the scheme of Section 4.2.2 and evaluated in an element of \(D\) in which the signed message is the user’ identity \(U_{i1}\). For the sake of simplicity, denote with \(G_{0}(x) = \sum_{i=0}^{\omega} x^{i}\) and with \(G_{1}(x, y_{1}, \ldots, y_{\omega-1}) = \sum_{i=0}^{\omega} \sum_{j=1}^{\omega-1} a_{ij} x^{i} y_{j}\), then we can represent \(U_{G,i1}\) as

\[
U_{G,i1} = G_{0}(d_{t}) + U_{i1} G_{1}(d_{t}, v_{1}).
\]

From Theorem 4.3 (Substitution attack case) we know that the success probability of finding a signature produced by TA on message \(U_{i1}\) is less than or equal to \(\frac{2}{q} - \frac{1}{q^{2}}\) and since the colluders can find the value \(d_{t} \in D\) (that was chosen randomly by the TA) with probability \(\frac{1}{q}\), we can conclude that the probability of finding a valid group identity of \(U_{i1}\) accepted by TA is

\[
Pr(U_{G,i1} \text{ is the group identity of } U_{i1}) \leq \frac{1}{q^{2}} \left( \frac{2}{q} - \frac{1}{q^{2}} \right).
\]

The value \(U_{G,i1}\) can be considered as a starting point to realize an impersonation attack as in Theorem 4.3. Hence, we can conclude that

\[
P_{G,I}^{TA} \leq \frac{1}{q^{2}} \left( \frac{2}{q} - \frac{1}{q^{2}} \right)^{2}.
\]
• Substitution attack for group signature: The proof for the substitution attack is similar to that of impersonation attack. The difference is that the success probability of the substitution attack of the scheme presented in Section 4.2.2 is less equal to \( \left( \frac{2}{q} - \frac{1}{q^2} \right) \), hence, we have
\[
P_{TA}^{G,S} \leq \frac{1}{q} q' \left( \frac{2}{q} - \frac{1}{q^2} \right)^2.
\]

• Transfer with a trap attack for group signature: Assume that after seeing a user signed message \((m_{i_0}, \alpha)\) published by \(U_{i_0}\), the colluders \(U_1, \ldots, U_{\omega-1}\) want to generate a group signature \((m_{i_0}, \alpha')\), such that \(\alpha' \neq \alpha\) and the user \(U_{i_1}\) and the TA will accept it as a valid group signed message of the user \(U_{i_0}\). Let \(\alpha = G(U_{i_1}, y_1, \ldots, y_{\omega-1}, m_{i_0})\). The first step that colluders can do is to find a valid group identity of user \(U_{i_0}\). To do this, they can compute a signature produced by TA on user's identity \(U_{i_0}\) using the scheme presented in Section 4.2.2 and evaluate it in an element \(d_t \in D\). Remember that we denote with \(U_{G,i_0}\) the group identity of user \(U_{i_0}\) and that we have defined in the case Impersonation attack for group signature \(G_0(x) = \sum_{i=0}^{\omega} x^i\) and \(G_1(x, y_1, \ldots, y_{\omega-1}) = \sum_{i=0}^{\omega} \sum_{j=1}^{\omega-1} a_{ij} x^i y^j\). To compute a TA's signature on user's identity \(U_{i_0}\), the colluders must realize a substitution attack on the scheme presented in Section 4.2.2. Based on Theorem 4.3 we know that the success probability of a substitution attack is less equal to \( \left( \frac{2}{q} - \frac{1}{q^2} \right) \) and since the value \(d_t\) is chosen randomly by the TA, the probability to find a value \(d'_t\) such that \(G_0(d'_t) + U_{i_0} G_1(d'_t, y_{TA}) = U_{G,i_0}\) is \(\frac{1}{q'}\). Then we can say that the probability of finding a group identity \(U_{G,i_0}\) of \(U_{i_0}\) is
\[
Pr(U_{G,i_0}) \leq \frac{1}{q'} \left( \frac{2}{q} - \frac{1}{q^2} \right).
\]
Given \(U_{G,i_0}\), the colluders must compute the signature \((m_{i_0}, \alpha'')\) such that \(\alpha'' = G(U_{G,i_0}, y_1, \ldots, y_{\omega-1}, m_{i_0})\). As before, the colluders can obtains the value \(\alpha''\) realizing an impersonation attack against a user that have as identity \(U_{G,i_0}\). The success probability of this attack is \(\frac{2}{q} - \frac{1}{q^2}\). Now, the pair \((m_{i_0}, \alpha'')\) can be consider as a starting point to begin a transfer with a trap attack of the original scheme presented in Section 4.2.2. Based on Theorem 4.3 we know that a transfer with a trap attack has a success probability of \(\frac{1}{q}\), then the \(P_{G,T}\) of our scheme is:
\[
P_{TA}^{G,T} \leq \frac{1}{q} q' \left( \frac{2}{q} - \frac{1}{q^2} \right)^2.
\]
• **Full-Anonymous**: To prove the Full-Anonymous property of the scheme, we calculate the probability of finding the identity of the signer from the group identity. Like to Theorem 5.3, the group identities are not used in user signatures and they are linearly independent, then $P_{TA}^{FA}(W, U_i)$ is

$$P_{TA}^{FA}(W, U_i) = Pr(W \text{ identifies } U_i | u_{G,i}).$$

As we have seen in Section 5.2.6 the group identity of user $U_\ell$ is obtained by TA in the following way:

$$U_{G,\ell} = \sum_{i=0}^{\omega} d_i^\ell + U_i \sum_{i=0}^{\omega-1} \sum_{j=1}^{\omega-1} a_{ij} d_i^\ell v_{TA,j}$$

where $v_{TA}$ is an element of $F_{q_0}^{\omega-1}$ used by TA for the verification of signatures, $d_i$ is an element of $\mathcal{D}$ chosen at random by TA and $(a_{ijk})$ are the coefficients of function $G(x, y, z)$ used to calculate the secret information of each user. Let indicate with $G_0(x) = \sum_{i=0}^{\omega} x^i$ and $G_1(x, y) = \sum_{i=0}^{\omega} \sum_{j=1}^{\omega-1} a_{ij} x^i y^j$ where $y \in F_{q_0}^{\omega-1}$ two polynomials of degree $\omega$. Then we can represent the group identity of $U_\ell$ as

$$U_{G,\ell} = G_0(d_\ell) + U_i G_1, (d_\ell, v_{TA}).$$

Since the polynomial $G_1(x, v_{TA})$ has degree $\omega$ then it has $\omega + 1$ coefficients, so the probability of finding $G_1(x, v_{TA})$ is

$$Pr(G_1(x, v_{TA}) | U_{G,\ell}) \leq \frac{\omega + 1}{q_0}.$$  

As we have seen in the Key Generation and Distribution by TA of Section 5.2.6, the TA chooses at random an element of $\mathcal{D}$ to calculate the group identity of a member of the group. Since $|\mathcal{D}| = q'$, the probability to find the value $d_\ell$ used for user $U_\ell$ is

$$Pr(d_\ell | U_{G,\ell}) = \frac{1}{q'}.$$  

Hence, the probability of recovering the user identity from group identity is

$$P_{FA}^{TA} = Pr(U_\ell | U_{G,\ell}) \leq \frac{\omega + 1}{q_0 q'}.$$  

• **Full-Traceability**: To prove the Full-Traceability property of the scheme we calculate the probability to create signature that cannot be traced back to any member of the coalition. Let consider the case in which a user $U_i$ generates a group signature
accepted by $U_j$ but such that in the Open a signature phase of Section 5.2.6 the TA cannot return a valid user identity. Let consider a group identity $U_{G,\ell}$ chose at random in $F_{q_0}$ and not used by any member of the coalition. The probability to find such group identity is

$$Pr(U_{G,\ell} \mid U_{G,\ell} \text{ is not used in } U) \leq 1 - \frac{n}{q_0}.$$ 

The coalition $W$ can do an impersonation attack against user $U_j$ such that $U_j$ accept the signature. From the Impersonation attack for user signature case, we know that the probability of such event is:

$$P_I = \frac{2}{q} - \frac{1}{q^2}$$

Since the group identity $U_{G,\ell}$ is not associated to any group member, then in the Open a signature phase the TA cannot return any valid user identity. Then

$$P_{TA} \leq \left(1 - \frac{n}{q_0}\right) \left(\frac{2}{q} - \frac{1}{q^2}\right).$$

Using the Definition 5.6 we have the proof.

\textbf{Theorem 5.7} In the above construction, the following modification also produces an $(n, \omega, \psi, 1 - \frac{n}{q_0}, 1 - \frac{n}{q_0})$-group full secure scheme: Instead of choosing randomly, the TA may choose $n$ elements $v_1, \ldots, v_n \in F_{q_0}^{\omega - 1}$, for user’s secret, such that for any $\omega$ vectors

$$v_{i_1} = (v_{1,i_1}, \ldots, v_{\omega-1,i_1}), \ldots, v_{i_\omega} = (v_{1,i_\omega}, \ldots, v_{\omega-1,i_\omega}),$$

the $\omega$ new vectors $(1, v_{1,i_1}, \ldots, v_{\omega-1,i_1}), \ldots, (1, v_{1,i_\omega}, \ldots, v_{\omega-1,i_\omega})$ are linearly independent.

\textbf{Proof.} The proof of this Theorem is similar to that of Theorem 5.6. The difference is in the use of Theorem 4.4.

Though the proposed $(n, \omega, \psi, 1 - \frac{n}{q_0}, 1 - \frac{n}{q_0})$-group full secure scheme is more secure than the proposed $(n, \omega, \psi, 2 - \frac{1}{q^2}, 1 - \frac{n}{q_0})$-group full secure scheme in term of impersonation or substitution, it requires more complicated transactions for generating each user’s secret information.
Corollary 5.1. The construction proposed in Theorem 5.7 is optimal in terms of the memory size of a signature.

The result holds because the required memory size for the above construction matches the lower bound on a signature given in Theorem 3.2.

5.2.7 Consideration on the security parameter $q'$

One of the parameters that guarantee the security of group signatures in both symmetric and asymmetric schemes, is the value $q'$ that is $|\mathcal{D}|$. Suppose that $q' = 1$, i.e., $\mathcal{D} = \{d_1\}$. As we have explained in symmetric and asymmetric schemes, the TA uses the elements of $|\mathcal{D}|$ to compute the group identity of each user. Since $|\mathcal{D}| = 1$, each group identity is obtained by using $d_1$. Let consider two users $U_i$ and $U_j$.

In symmetric construction each user has as group identity $u_{G,i}$ and $u_{G,j}$ respectively defined as:

$$u_{G,\ell} = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,\ell-1}w_{TA,i}d_i^\ell + u_\ell \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,\ell}w_{TA,j}d_j^\ell.$$

for $\ell = i, j$. Let define $F_0(x, y) = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,\ell-1}x^iy^j$ and $F_1(x, y) = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} a_{i,j,\ell}x^iy^j$. If $U_i$ and $U_j$ put together their group identities they obtain the system

$$\begin{cases}
u_{G,i} = F_0(u_{TA}, d_1) + u_i F_1(u_{TA}, d_1) \\
u_{G,j} = F_0(u_{TA}, d_1) + u_j F_1(u_{TA}, d_1)
\end{cases}$$

that they use to find the values $F_0(u_{TA}, d_1)$ and $F_1(u_{TA}, d_1)$, that is the values used by TA to compute the group identity of every user. The values $F_0(u_{TA}, d_1)$ and $F_1(u_{TA}, d_1)$ can be used by $U_i$ and $U_j$ to forge false user identities for symmetric scheme accepted by each member of the group.

In the asymmetric construction $U_i$ and $U_j$ receive as group identities the values $U_{G,i}$ and $U_{G,j}$ defined as:

$$U_{G,\ell} = \sum_{i=0}^{\omega} d_i^\ell + u_\ell \sum_{i=0}^{\omega} \sum_{j=1}^{\omega-1} a_{i,j,\ell}d_i^\ell v_{TA,j}\ell$$

for $\ell = i, j$. As before, let define $G_0(x) = \sum_{i=0}^{\omega} x^i$ and $G_1(x, y) = \sum_{i=0}^{\omega} \sum_{j=1}^{\omega-1} a_{i,j}x^iy_j$ where $y \in F_{q_0}^{\omega-1}$. If $U_i$ and $U_j$ put together their group identities, they can resolve the system

$$\begin{cases}
u_{G,i} = G_0(d_1) + U_i G_1(d_1, v_{TA}) \\
u_{G,j} = G_0(d_1) + U_j G_1(d_1, v_{TA})
\end{cases}$$
and found the values $G_0(d_1)$ and $G_1(d_1, v_{TA})$ used by TA to compute group identities. In this way $U_i$ and $U_j$ can forge false user group identities in the asymmetric scheme, accepted by each member of the group. This can be seen as a possible attack against both symmetric and asymmetric group signature schemes.

5.3 Unconditionally Secure Group Signature opened by the signer

In the schemes presented in Section 5.2, we have described some Unconditionally Secure Group Signature schemes, in which the TA distributes user and group identities used by each member of the group to generate a user or a group signature. The group identity of each user is obtained as a signature of the identity of the user and each time a group signature must be open the group identity is sent to the TA that determines the identity of the signer.

In the following we will define a new scheme referred to as Unconditionally Secure Group Signature opened by the signer, in which as before, each user can generate a user or a group signature using the identities distributed by TA, but unlike schemes of Section 5.2, in the open phase only the signer can reveal some information that can be used by each member of the group to verify that he has generated the group signature.

5.3.1 The model

In the following we give a formal definition of Unconditionally Secure Group Signature opened by the signer:

Definition 5.7 A scheme $\Pi$ is an Unconditionally Secure Group Signature opened by the signer if it is constructed as follows:

1. Notation: $\Pi$ consists of $(TA, U, M, S, V, A, \text{Sig, Ver, Open})$ where

   - TA is a trusted authority,
   - $U$ is a finite set of users,
   - $M$ is a finite set of possible messages,
   - $S$ is a finite set of possible signing-keys,
• $\mathcal{V}$ is a finite set of possible verification-keys,
• $\mathcal{A}$ is a finite set of possible signatures,
• $\text{Sig}: \mathcal{S} \times \mathcal{M} \to \mathcal{A}$ is a signing algorithm,
• $\text{Ver}: \mathcal{M} \times \mathcal{A} \times \mathcal{V} \times \{\mathcal{U} \cup \emptyset\} \to \{\text{accept, reject}\}$ is a verification algorithm,
• $\text{Open}: \mathcal{A} \times \mathcal{U} \to \{\text{true, false}\}$ is an algorithm that is used to verify that a given group identity is associated to a member of the group.

2. Key Pair Generation and Distribution by TA: For each user $U_i \in \mathcal{U}$ ($1 \leq i \leq n$), the TA chooses a signing key $s_i \in \mathcal{S}$ and a verification-key $v_i \in \mathcal{V}$, and transmits the pair $(s_i, v_i)$ to $U_i$ via a secure channel. After delivering these keys, the TA may erase the pair $(s_i, v_i)$ from his memory and each user keeps his secret information secret.

3. Signature Generation: For a message $m \in \mathcal{M}$, a user $U_i$ generate a signature $\alpha = \text{Sig}(s_i, m) \in \mathcal{A}$ by using his/her secret information in conjunction with the signing algorithm. The pair $(m, \alpha)$ is regarded as:
   • a signed message of $U_i$,
   • a group signed message that is a signature generated by a member of the group using $U_{G,i}$ as his group identity.

4. Signature Verification: On receiving $(m, \alpha)$, the user $U_j$ checks if the signature is a user or a group signature. If $(m, \alpha)$ is a user signature then $U_j$ accepts it as valid signature from $U_i$ if $\text{Ver}(m, \alpha, e_j, U_i) = \text{accept}$, else if $(m, \alpha)$ is a group signature then $U_j$ accept it as valid if $\text{Ver}(m, \alpha, e_j, U_{G,i}) = \text{accept}$. Of course if a member of a group know $U_{G,i}$, he cannot determine the identity of the signer.

5. Open a signature: On receiving the pair $(U_{G,i}, U_i)$ from user $U_i$, each member of the group can verify that the author of group signature is $U_i$.

The main difference with the scheme presented in Section 5.2.1 is in the Open a signature phase. Indeed, in Section 5.2.1 only the TA can open a signature and reveals the real identity of the signer, while in this new scheme the signer himself must reveals his identity and give to the other members of the group the proof that he has generated the signature. The reason for which we call this scheme opened by the signer is that only the
Chapter 5. Unconditionally Secure Group Signatures

signer has the information with which the signature’s originator can be found, while the TA is used only during the Key Pair Generation and Distribution by TA phase and not in the other phases.

5.3.2 Properties and attacks

As for the scheme opened by TA we give now the definitions of attacks and properties of the new schemes opened by the signer.

Definition 5.8 The success probabilities of impersonation, substitution and transfer with a trap attacks in a group signature scheme opened by the signer, denoted by \( P_{sgn}^{G,I} \), \( P_{sgn}^{G,S} \) and \( P_{sgn}^{G,T} \) respectively, are formally defined as follows:

1. Success probability of Impersonation: for \( W \in \mathcal{W} \) and \( U_i, U_j \in \mathcal{U} \) with \( U_i, U_j \not\in W \), we define \( P_{sgn}^{G,I}(U_i, U_j, W) \) as:

\[
P_{sgn}^{G,I}(U_i, U_j, W) = \max_{sW, vW} \max_{1 \leq k \leq n, k \neq i} \max_{\left\{(m_{k,\ell}, \alpha_{k,\ell})\right\}} \max_{(m, \alpha) \in \mathcal{U}} Pr(U_j \text{ traced } u_{G,i} \text{ to user } U_i)
\]

\[
Pr(U_j \text{ accepts } (m, \alpha) \text{ as a valid group signature} | sW, vW, \left\{c_k\right\})
\]

where \( u_{G,i} \) is the group identity of \( U_i \), \( c_k = \left\{(m_{k,\ell}, \alpha_{k,\ell})\right\} \) is taken over a family of possible sets of valid user signed messages generated by a user \( U_k \) (1 \( \leq k \leq n, k \neq i \)) such that \( 0 \leq |c_k| \leq \psi \) (1 \( \leq k \leq n, k \neq i \)). Note that \( m_{k,\ell} \) are not necessarily distinct. Then, \( P_{sgn}^{G,I} \) is given as \( P_{sgn}^{G,I} = \max_{U_i, U_j, W} P_{sgn}^{G,I}(U_i, U_j, W) \), where \( W \in \mathcal{W} \) and \( U_i, U_j \in \mathcal{U} \) with \( U_i, U_j \not\in W \).

2. Success probability of Substitution: for \( W \in \mathcal{W} \) and \( U_i, U_j \in \mathcal{U} \) with \( U_i, U_j \not\in W \), we define \( P_{sgn}^{G,S}(U_i, U_j, W) \) as

\[
P_{sgn}^{G,S}(U_i, U_j, W) = \max_{sW, vW} \max_{1 \leq k \leq n} \max_{c_k = \left\{(m_{k,\ell}, \alpha_{k,\ell})\right\}} \max_{(m, \alpha) \in \mathcal{U}} Pr(U_j \text{ traced } u_{G,i} \text{ to user } U_i)
\]

\[
Pr(U_j \text{ accepts } (m, \alpha) \text{ as a valid group signature} | sW, vW, \left\{c_k\right\})
\]

where \( u_{G,i} \) is the group identity of \( U_i \), \( c_k = \left\{(m_{k,\ell}, \alpha_{k,\ell})\right\} \) is taken over a family of possible sets of valid signed messages generated by \( U_k \) (1 \( \leq k \leq n \)) such that \( 0 < |c_i| \leq \psi \) and \( 0 \leq |c_k| \leq \psi \) (1 \( \leq k \leq n, k \neq i \)) and \( (m, \alpha) \) is taken such that
Chapter 5. Unconditionally Secure Group Signatures

80 Chapter 5. Unconditionally Secure Group Signatures

respectively, are formally defined as follows:

The success probabilities of an attack against Full-Anonymity and Full-Traceability in a group signature scheme opened by the signer, denoted by $P_{FT}$ and $P_{GA}$ respectively, are formally defined as follows:

1. **Probability of Full-Anonymity:** for $W \in \mathcal{W}$ and $U_i \in \mathcal{U}$ with $U_i \notin W$, we define $P_{GA}(W, U_i)$ as:

\[
P_{GA}(W, U_i) = \max_{s_W, v_W} \max_{1 \leq k \leq n} \max_{\{m_k, \alpha\}} \max_{u_G, i} \max_{\{c_k\}} \Pr(U_i \text{ identifies } u_{G, i}, s_W, v_W, \{c_k\}, GId),
\]

where $u_{G, i}$ is the group identity of user $U_i$, $c_k = \{(m_k, \alpha_k)\}$ is taken over a family of possible sets of valid signed messages generated by $U_k$ ($1 \leq k \leq n$, $k \neq i$) such that $0 \leq |c_k| \leq \psi$ ($1 \leq k \leq n$, $k \neq i$), $(m, \alpha)$ is taken over a family of possible signed messages generated by $U_i$ and $\alpha'$ is taken such that $\alpha \neq \alpha'$. Then $P_{GA}$ is given as $P_{GA} = \max_{W, U_i} P_{GA}(W, U_i)$.

2. **Probability of Full-Traceability:** for $W \in \mathcal{W}$ and $U_i \in \mathcal{U}$ with $U_i \notin W$, we define $P_{FT}(U_i, W)$ as:

\[
P_{FT}(U_i, W) = \max_{s_W, v_W} \max_{1 \leq k \leq n} \max_{\{m_k, \alpha\}} \max_{u_G, i} \max_{\{c_k\}} \Pr(U_i \text{ traces } u_{G, i} \text{ to user } U_i)
\]

where $u_{G, i}$ is the group identity of user $U_i$, $c_k = \{(m_k, \alpha_k)\}$ is taken over a family of possible sets of valid signed messages generated by $U_k$ ($1 \leq k \leq n$, $k \neq i$) such that $0 \leq |c_k| \leq \psi$ ($1 \leq k \leq n$) and $GId$ is a set of group identities known by $W$. Then, $P_{FT}$ is given as $P_{FT} = \max_{W} P_{FT}(W, U_i)$.

$P_{GA}$ and $P_{FT}$ are given as:

\[
P_{GA} = \max_{U_i, U_j, W} P_{GA}(U_i, U_j, W)
\]

and

\[
P_{FT} = \max_{U_i, U_j, W} P_{FT}(U_i, U_j, W)
\]

where $U_i, U_j \in \mathcal{U}$ with $U_i, U_j \notin W$. Note that $m_i, j$ are not necessarily distinct. Then, $P_{GA}$ is given as $P_{GA} = \max_{U_i, U_j, W} P_{GA}(U_i, U_j, W)$, where $W \in \mathcal{W}$ and $U_i, U_j \in \mathcal{U}$ with $U_i, U_j \notin W$. 

3. **Success probability of Transfer with a trap:** for $W \in \mathcal{W}$ and $U_i, U_j \in \mathcal{U}$ with $U_j \notin W$ we define $P_{GA}(U_i, U_j, W)$ as

\[
P_{GA}(U_i, U_j, W) = \max_{s_W, v_W} \max_{1 \leq k \leq n} \max_{\{m_k, \alpha\}} \max_{u_G, i} \max_{\{c_k\}} \Pr(U_i \text{ accepts } (m, \alpha) \text{ as a valid group signature } |s_W, v_W, (m, \alpha))
\]

where $u_{G, i}$ is the group identity of $U_i$, $c_k = \{(m_k, \alpha_k)\}$ is taken over a family of possible sets of valid signed messages generated by $U_k$ ($1 \leq k \leq n$, $k \neq i$) such that $0 \leq |c_k| \leq \psi$ ($1 \leq k \leq n$, $k \neq i$), $(m, \alpha)$ is taken over a family of possible signed messages generated by $U_i$ and $\alpha'$ is taken such that $\alpha \neq \alpha'$. Then $P_{GA}$ is given as $P_{GA} = \max_{U_i, U_j, W} P_{GA}(U_i, U_j, W)$ where $W \in \mathcal{W}$ and $U_i, U_j \in \mathcal{U}$ with $U_j \notin W$. 

Definition 5.9 The success probabilities of an attack against Full-Anonymity and Full-Traceability in a group signature scheme opened by the signer, denoted by $P_{FT}$ and $P_{GA}$ respectively, are formally defined as follows:

1. **Probability of Full-Anonymity:** for $W \in \mathcal{W}$ and $U_i \in \mathcal{U}$ with $U_i \notin W$, we define $P_{GA}(W, U_i)$ as:

\[
P_{GA}(W, U_i) = \max_{s_W, v_W} \max_{1 \leq k \leq n} \max_{\{m_k, \alpha\}} \max_{u_G, i} \max_{\{c_k\}} \Pr(U_i \text{ identifies } u_{G, i}, s_W, v_W, \{c_k\}, GId),
\]

where $u_{G, i}$ is the group identity of user $U_i$, $c_k = \{(m_k, \alpha_k)\}$ is taken over a family of possible sets of valid signed messages generated by $U_k$ ($1 \leq k \leq n$) such that $0 \leq |c_k| \leq \psi$ ($1 \leq k \leq n$) and $GId$ is a set of group identities known by $W$. Then, $P_{GA}$ is given as $P_{GA} = \max_{W} P_{GA}(W, U_i)$.

2. **Probability of Full-Traceability:** for $W \in \mathcal{W}$ and $U_i \in \mathcal{U}$ with $U_i \notin W$, we define $P_{FT}(U_i, W)$ as:

\[
P_{FT}(U_i, W) = \max_{s_W, v_W} \max_{1 \leq k \leq n} \max_{\{m_k, \alpha\}} \max_{u_G, i} \max_{\{c_k\}} \Pr(U_i \text{ traces } u_{G, i} \text{ to user } U_i)
\]
\[ \Pr(U_j \text{ accepts } (m, \alpha) \text{ as valid group signature}| s_W, v_W, \{c_k\}, GId) \]
\[ \Pr(U_j \text{ cannot trace back to any member of } \mathcal{U}) \]

where \( GId \) is a set of group identities know by \( W \) and \( c_k = \{(m_{k,\ell}, \alpha_{k,\ell})\} \) is taken over a family of possible sets of valid user signed messages generated by a user \( U_k \) (\( 1 \leq k \leq n, k \neq i \)) such that \( 0 \leq |c_k| \leq \psi \) (\( 1 \leq k \leq n, k \neq i \)). Note that \( m_{k,\ell} \) are not necessarily distinct. Then, \( P_{FT}^{\text{sgn}} \) is given as \( P_{FT}^{\text{sgn}} = \max_{U_i, W} P_{FT}^{\text{sgn}}(U_i, W) \), where \( W \in W \) and \( U_i \in \mathcal{U} \) with \( U_i \notin W \).

The concept of \((n, \omega, \psi, p_1, p_2)\)-group secure and of \((n, \omega, \psi, p_1, p_2, p_3, p_4)\)-group full secure signature scheme can now be defined, where \( p_1, p_2, p_3 \) and \( p_4 \) are security parameters whose meanings will be made precise in the following definitions.

**Definition 5.10** Let \( \Pi \) be an Unconditionally Secure Group Signature Scheme opened by the signer. Let \( \mathcal{U} \) be a set of \( n \) users. Then, \( \Pi \) is \((n, \omega, \psi, p_1, p_2)\)-group secure if the following conditions are satisfied: as long as there exist at most \( \omega - 1 \) colluders and each user is allowed to generate at most \( \psi \) signatures, the following inequalities hold:

\[ \max\{P_I, P_{G,I}^{\text{sgn}}, P_S, P_{G,S}^{\text{sgn}}\} \leq p_1, \quad \max\{P_T, P_{G,T}^{\text{sgn}}\} \leq p_2 \]

where \( P_I \) and \( P_{G,I}^{\text{sgn}} \) are the probabilities of success in impersonation for user and group signatures respectively, \( P_S \) and \( P_{G,S}^{\text{sgn}} \) are the probabilities of success in substitution for user and group signatures respectively and \( P_T \) and \( P_{G,T}^{\text{sgn}} \) are the probabilities of success in transfer with a trap for user and group signatures respectively.

As before, we note that there is an alternative definition of security in which one may use a single security parameter \( p \) instead and define the success probability as

\[ \max\{P_I, P_{G,I}^{\text{sgn}}, P_S, P_{G,S}^{\text{sgn}}, P_T, P_{G,T}^{\text{sgn}}\} \leq p. \]

In practice, however, some applications may attach more weight to strength against impersonation and substitution than against transfer with a trap, while some other applications may have an emphasis on robustness against transfer with a trap. By introducing two separate parameters \( p_1 \) and \( p_2 \), we have an opportunity to design a signature scheme with fine-tuned level of security.
Definition 5.11 Let \( \Pi \) be an Unconditionally Secure Group Signature Scheme opened by the signer. Let \( \mathcal{U} \) be a set of \( n \) users. Then, \( \Pi \) is \((n, \omega, \psi, p_1, p_2, p_3, p_4)\)-group full secure if it is \((n, \omega, \psi, p_1, p_2)\)-group secure and the following inequalities holds:

\[
P^{\text{sgn}}_{FA} \leq p_3, \quad P^{\text{sgn}}_{FT} \leq p_4
\]

where \( P^{\text{sgn}}_{FA} \) and \( P^{\text{sgn}}_{FT} \) are the probabilities of attacks against Full-Anonymity and Full-Traceability properties respectively.

As before, an alternative definition can be used in substitution of Definition 5.11 in which is used a single parameter. Now we prefer to introduce \( p_3 \) and \( p_4 \) to give evidence to Full-Anonymity and Full-Traceability properties of a signature scheme.

5.3.3 A symmetric construction

In this Section we show an implementation of Unconditionally Secure Group Signature opened by the signer referred to as symmetric construction. In this construction, we will use the same function both of signing and of verifying and then \( S = \mathcal{V} \).

1. Key Generation and Distribution by TA: Let \( F_{q_0} \) be a finite field with \( q_0 \) elements such that \( q_0 \geq n\omega q \), where \( q \) is a security parameter of the system. We assume that the size of \( q_0 \) is almost the same as \( n\omega q \). Then TA divides \( F_{q_0} \) into \( n \) disjoint subsets \( \mathcal{U}_1, \ldots, \mathcal{U}_n \), such that \( |\mathcal{U}_\ell| = \omega q \) for any \( \ell \). Here the subsets \( \mathcal{U}_\ell \) \((\ell = 1 \ldots n)\) are made public for all users. The TA picks uniformly at random a value \( u_{TA} \) in \( F_{q_0} \) and constructs uniformly at random a matrix \( A = (a_{i,j,k}) \) \((0 \leq i \leq \omega, 0 \leq j \leq \omega, 0 \leq k \leq \psi)\) such that for any fixed \( k \) the matrix \( A_k = (a_{i,j}) \) is symmetric. Given \( A \), the TA defines the polynomial used to create the secret information for each user as

\[
F(x, y, z) = \sum_{i=0}^{\omega} \sum_{j=0}^{\omega} \sum_{k=0}^{\psi} a_{i,j,k} x^i y^j z^k.
\]

For each user \( \mathcal{U}_\ell \) \((1 \leq \ell \leq n)\) the TA picks uniformly at random a value \( u_\ell \) in \( \mathcal{U}_\ell \) and calculates

\[
U_{G,\ell}(x) = F(x, u_{TA}, u_\ell)
\]

and

\[
u_{G,\ell} = U_{G,\ell}(u_\ell).
\]
The value $u_\ell$ represents the user identity, while $U_{G,\ell}(x)$ is obtained as a signature of $u_\ell$ generated by TA and used in the Open phase as proof of the signer. Moreover, we assume that each message $m$ is an element in $F_{q_0}$ as well. For each user $U_\ell$ ($1 \leq \ell \leq n$), the TA computes his secret information

$$e_\ell = \{F(x, u_\ell u_{G,\ell}, z), u_\ell, U_{G,\ell}(x)\}$$

and sends it to $U_\ell$ over a secure channel. When $U_\ell$ receive $e_\ell$, he can verify if $U_{G,\ell}(x)$ is a correct signature of $u_\ell$ generated by TA. Once the secret information has been delivered, there is now no need for the TA to keep it.

2. Signature Generation: For a message $m \in F_{q_0}$, the user $U_s$ can choose to send a user or a group signature:

- If $U_s$ wants to generate a user signature then he computes $u_{G,s}$ and $F(x, u_s, z)$ using his secret information in the following way

  $$u_{G,s} = U_{G,s}(u_s)$$

  and

  $$F(x, u_s, z) = F(x, u_s u_{G,s}, z) \oplus (1, 1/u_{G,s}, 1).$$

  The signature of message $m$ is $\alpha = \{a_{s,m}(x), u_s\}$ where $a_{s,m}(x) = F(x, u_s, z)\big|_{z=m}$.

  Then, $U_s$ sent the pair $(m, \alpha)$ including his identity.

- Otherwise, if $U_s$ wants to generate a group signature, then he computes $F(x, u_{G,s}, z)$ using his secret information in the following way

  $$F(x, u_{G,s}, z) = F(x, u_s u_{G,s}, z) \oplus (1, 1/u_s, 1).$$

  The signature of message $m$ is $\alpha = \{a_{s,m}(x), u_{G,s}\}$ where $a_{s,m}(x) = F(x, u_{G,s}, z)\big|_{z=m}$.

  Then, $U_s$ sent the pair $(m, \alpha)$ without including his identity.

3. Signature Verification: On receiving a pair $(m, \alpha)$, the user $U_r$ checks if it is a user or a group signature. This can be seen by checking the presence of the signer’s identity. By using his secret information $e_r$, $U_r$ checks whether $\alpha$ is valid or not.

More precisely, let $\bar{u}_s$ defined as:

$$\bar{u}_s = \begin{cases} 
  u_s & \text{if } (m, \alpha) \text{ is a user signature} \\
  u_{G,s} & \text{if } (m, \alpha) \text{ is a group signature}
\end{cases}$$
then $U_s$ accepts $(m, \alpha)$ as being a valid message-signature pair if

$$F(x, u_ru_G,r, z)|_{x=\bar{u}_s, z=m} = a_{s,m}(x)|_{x=ur_G,r}$$

and in the case of a user signature, $U_r$ checks also that $u_r \in U_r$.

4. **Open a signature**: Once a group signature’s generator wants to reveal his identity, he can send the pair $(U_{G,s}(x), u_s)$ and each group member verify the signature of $u_s$ generated by TA. More precisely, when user $U_r$ receive the pair $(U_{G,s}(x), u_s)$, he checks if $U_{G,s}(u_ru_G,r) = F(x, u_ru_G,r, u_s)|_{x=u_{TA}}$ and if $U_{G,s}(u_s) = u_{G,s}$. Remember that $U_r$ has received $u_{G,s}$ together with a group signature.

**Theorem 5.8** The required memory size in the above construction is given as follows:

$$|S| = |V| = \omega q_0^{(\omega+1)(\psi+2)}$$

$$|A| = \begin{cases} 
\omega q_0^{\omega+1} & \text{for a user signature} \\
q_0^{\omega+2} & \text{for a group signature.} 
\end{cases}$$

**Theorem 5.9** The above symmetric signature scheme results in a $(n, \omega, \frac{1}{q_0}, \frac{1}{q_0}, \frac{1}{q_0}, \left(1 - \frac{n}{q_0}\right) \frac{1}{q_0})$-full group secure scheme.

**Proof.** The proof is similar to that of Theorem 5.3 except for the $P_{FA}^{\text{sgn}}$ probability. $P_{FA}^{\text{sgn}}$ is the probability of finding the identity of the signer from the group identity. In the presented construction, such probability is like to choose randomly an element in $F_{q_0}$. Then

$$P_{FA}^{\text{sgn}} = Pr(u_i \in R F_{q_0}) = \frac{1}{q_0}. $$

**5.3.4 An asymmetric construction**

In this Section, we propose an asymmetric construction of Unconditionally Secure Group Signature opened by the signer scheme in which the secret information for signing and for verifying are different.
1. **Key Pair Generation and Distribution by TA:** Let $F_q$ be a finite field with $q$ elements such that $q \geq n$. The TA picks $n + 1$ elements $v_{TA}, v_1, v_2, \ldots, v_n$ uniformly at random in $F_q^{\omega-1}$ for himself and for users $U_1, U_2, \ldots, U_n$ respectively and constructs a polynomial $G(x, y_1, \ldots, y_{\omega-1}, z)$ as follows:

$$G(x, y_1, \ldots, y_{\omega-1}, z) = \sum_{i=0}^{\omega} \sum_{k=0}^{\psi} a_{i0k}x^i z^k + \sum_{i=0}^{\omega-1} \sum_{j=1}^{\omega} \sum_{k=0}^{\psi} a_{ijk}x^i y_j z^k$$

where the coefficients $a_{ijk}$ ($0 \leq i \leq \omega$, $1 \leq j \leq \omega - 1$ and $0 \leq k \leq \psi$) are chosen uniformly at random from $F_q$. Moreover, we assume that each user’s identity $U_\ell$ and message $m$ are elements of $F_q$. For each user $U_\ell$ ($1 \leq \ell \leq n$), the TA computes the group identity as:

$$U_{G,\ell}(y_1, \ldots, y_{\omega-1}) = G(x, y_1, \ldots, y_{\omega-1}, U_\ell)|_{x=U_{TA}}$$

and

$$U_{G,\ell} = U_{G,\ell}(U_\ell, \ldots, U_{\ell}^{\omega-1})$$

that is a signature produced by TA and valued in the user identity $U_\ell$ and computes the signing and verification key as:

$$s_\ell = G(U_\ell U_{G,\ell}, y_1, \ldots, y_{\omega-1}, z)$$

$$\tilde{v}_\ell = G(x, v_\ell, z).$$

Then $U_\ell$’s secret information is

$$e_\ell = \{s_\ell, v_\ell, \tilde{v}_\ell, U_{G,\ell}(y_1, \ldots, y_{\omega-1})\}$$

that the TA sends to $U_\ell$ over a secure channel. When $U_\ell$ receives $e_\ell$, he verifies if $U_{G,\ell}(y_1, \ldots, y_{\omega-1})$ is a correct signature of $U_\ell$ generated by TA. Once all the keys are delivered, there is no need for the TA to keep users’ secret information.

2. **Signature Generation:** For a message $m \in F_q$, the user $U_{\ell_1}$ can generate both a user or a group signature. There are two cases:

- if $U_{\ell_1}$ wants to generate a user signature then he computes $U_{G,\ell_1}$ and $G(U_{\ell_1}, y_1, \ldots, y_{\omega-1}, z)$ as:

$$U_{G,\ell_1} = U_{G,\ell_1}(U_{\ell_1}, \ldots, U_{\ell_1}^{\omega-1})$$
and
\[ G(U_{\ell_1}, y_1, \ldots, y_{\omega-1}, z) = G(U_{\ell_1}, y_1, \ldots, y_{\omega-1}, z) \oplus (1/U_{G,\ell_1}, 1, \ldots, 1, 1). \]

The signature of message \( m \) is
\[ \alpha = G(U_{\ell_1}, y_1, \ldots, y_{\omega-1}, z)|_{z=m} = G(U_{\ell_1}, y_1, \ldots, y_{\omega-1}, m) \]
and the tuple \((m, \alpha, U_{\ell_1})\) is sent by \( U_{\ell_1} \).

• otherwise, if \( U_{\ell_1} \) wants to generate a group signature then using his signing key \( s_{\ell_1} \), he computes
\[ G(U_{G,\ell_1}, y_1, \ldots, y_{\omega-1}, z) = G(U_{\ell_1}U_{G,\ell_1}, y_1, \ldots, y_{\omega-1}, z) \oplus (1/U_{\ell_1}, 1, \ldots, 1, 1). \]

The signature of message \( m \) is
\[ \alpha = G(U_{G,\ell_1}, y_1, \ldots, y_{\omega-1}, z)|_{z=m} = G(U_{G,\ell_1}, y_1, \ldots, y_{\omega-1}, m) \]
and the tuple \((m, \alpha, U_{G,\ell_1})\) is sent by \( U_{\ell_1} \).

3. **Signature Verification**: On receiving the pair \((m, \alpha)\), the user \( U_{\ell_2} \) checks if it is a user or a group signature. As before, this can be seen by checking the presence of the signer’s identity. By using his verification key \( \tilde{\nu}_{\ell_2} \), \( U_{\ell_2} \) checks whether \( \alpha \) is valid or not. More precisely, let
\[ \tilde{U}_{\ell_1} = \begin{cases} U_{\ell_1} & \text{if } (m, \alpha) \text{ is a user signature} \\ U_{G,\ell_1} & \text{if } (m, \alpha) \text{ is a group signature} \end{cases} \]
\( U_{\ell_2} \) accept \((m, \alpha)\) as being a valid message-signature pair if
\[ \tilde{\nu}_{\ell_2}|_{x=\tilde{U}_{\ell_1}, z=m} = G(x, \nu_{\ell_2}, z)|_{x=\tilde{U}_{\ell_1}, z=m} = \alpha|_{(y_1, y_2, \ldots, y_{\omega-1})=(v_1, v_2, \ldots, v_{\omega-1}, v_2)}. \]

4. **Open a signature**: Once a group signature’s generator wants to reveal his identity, he can send the pair \((U_{G,\ell_1}(y_1, \ldots, y_{\omega-1}), U_{\ell_1})\) and each group members verifies the signature of \( U_{\ell_1} \) generated by TA. More precisely, when user \( U_{\ell_2} \) receives the pair \((U_{G,\ell_1}(y_1, \ldots, y_{\omega-1}), U_{\ell_1})\), he checks if \( U_{G,\ell_1}(\nu_{\ell_2}) = G(x, \nu_{\ell_2}, U_{\ell_1})|_{x=U_{TA}} \) and if \( U_{G,\ell_1}(U_{\ell_1}, \ldots, U_{\ell_1}^{\omega-1}) = U_{G,\ell_1} \). Remember that \( U_{\ell_2} \) has received \( U_{G,\ell_1} \) together with a group signature.
Theorem 5.10 The required memory size in the above construction is given as follows:

\[ |S| = q^{\omega(\psi+2)-1} \]
\[ |V| = q^{2(\omega+1)(\psi+2)-2} \]
\[ |A| = q^{\omega+1} \]

Theorem 5.11 The above asymmetric scheme for group signatures results in a \((n, \omega, \frac{2}{q}, \frac{1}{q^2}, \frac{1}{q^2}, \frac{1}{q^2}, \frac{1}{q^2})\) - full group secure scheme.

Proof. The proof is similar to that of Theorem 5.6 except for the \(P_{FA}^{\text{sgn}}\) probability. \(P_{FA}^{\text{sgn}}\) is the probability to find the identity of the signer from the group identity. In the presented construction, such probability is like to choose randomly an element in \(F_{q^0}\). Then

\[ P_{FA}^{\text{sgn}} = \Pr(u_i \in R F_{q^0}) = \frac{1}{q^0} \]

Theorem 5.12 In the above construction, the following modification also produces an \((n, \omega, \psi, \frac{1}{q}, \frac{1}{q}, \frac{1}{q}, \frac{1}{q}, \left(1 - \frac{n}{q^0}\right) \left(\frac{2}{q} - \frac{1}{q^2}\right))\) - group full secure scheme: Instead of choosing randomly, the TA may choose \(n\) elements \(v_1, \ldots, v_n \in F_{q^0}^{-1}\), for user’s secret, such that for any \(\omega\) vectors

\[ v_1 = (v_{1,i_1}, \ldots, v_{\omega-1,i_1}), \ldots, v_\omega = (v_{1,i_\omega}, \ldots, v_{\omega-1,i_\omega}), \]

the \(\omega\) new vectors \((1, v_{1,i_1}, \ldots, v_{\omega-1,i_1}), \ldots, (1, v_{1,i_\omega}, \ldots, v_{\omega-1,i_\omega})\) are linearly independent.

Proof. The proof is like that of Theorem 5.7.

Though the proposed \((n, \omega, \psi, \frac{1}{q}, \frac{1}{q}, \frac{1}{q}, \frac{1}{q}, \left(1 - \frac{n}{q^0}\right) \left(\frac{2}{q} - \frac{1}{q^2}\right))\) - group full secure scheme is more secure than the proposed \((n, \omega, \psi, \frac{2}{q}, \frac{1}{q}, \frac{1}{q}, \frac{1}{q}, \left(1 - \frac{n}{q^0}\right) \left(\frac{2}{q} - \frac{1}{q^2}\right))\)-group full secure scheme in term of impersonation or substitution, it requires more complicated transactions for generating each user’s secret information.

Corollary 5.2 The construction proposed in Theorem 5.12 is optimal in terms of the memory size of a signature.

The results holds because the required memory size for the above construction matches the lower bound on a signature given by Theorem 3.2.
Chapter 6

Conclusions

In this thesis we have studied signature and group signature schemes in the unconditionally secure setting. We have introduced the concept of group signature in such a setting, by extending the definition of unconditionally secure signatures and we have provided some constructions as well.

Digital signatures are an important technology for ensuring unforgeability and non-repudiation of digital data. Chaum and van Heyst [16] have extended classical signature schemes introducing a new cryptographic concept, referred to as group signature schemes. In contrast with ordinary signatures, group signatures provide anonymity of the signer, i.e., a verify can only tell that a member of some group signed. Moreover, in exceptional cases such as a legal dispute, any group signature can be ”opened” by a designed Trusted Authority to reveal unambiguously the identity of the signature’s originator. As we have seen in Section 1.3, these schemes rely their security on the presumed computational difficulty of solving certain number theoretic problems (e.g. factoring large composites or computing discrete logarithms in large finite fields). But progresses in computer and refinements of various algorithms have made possible to solve large instances of the aforementioned problems (see Table 1.3 and [13]) providing no confidence on the long term integrity of data signed by these schemes. For this reason, it has been necessary to define digital signature schemes that provide assurance of long term integrity.

Some constructions of unconditionally secure schemes can be obtained by using Multi-receiver Authentication Codes (MRA), in which a sender wishes to authenticate a message for a group of receivers, such that each receiver can verify authenticity of the received message. MRA codes are described in Chapter 2, where we give the model, the definition of
some attacks such as impersonation and substitution, and some bounds on the success probability of the attacks and on the size of some elements of the model [55]. Among all the constructions, the DFY polynomial construction [25] reaches these bounds by equality and can be considered as a starting point for the definition of other MRA codes: MRA-code for multiple message transmissions and MRA-codes with dynamic sender (referred to as DMRA). A MRA-code for multiple messages transmissions allows a sender to sign more than one message, while in a DMRA-code everyone in the group can be the signer. Both schemes are described in Chapter 3. The constructions we have described are a generalization of DFY scheme.

In all the described MRA schemes, messages are transmitted over a broadcast channel, hence, transferability, i.e. the possibility a user has to forward securely a signed message to another user who can still verify the signature, is not required. If a point-to-point network is used, then a new attack, called transfer with a trap, is possible: a user can modify a received message in such a way that a third part can accept it as authentic. For a digital signature, transferability is a property that cannot be neglected. Unconditionally secure digital signature schemes are dynamic MRA-codes with transferability, i.e. codes that permit an unlimited transfer of a signature without compromising the security of the scheme. In Chapter 4, we describe two new models for unconditionally secure digital signature schemes: the first is basically a DMRA code for a single message, while the second permits multiple message transmission.

Unconditionally secure digital signature schemes can be useful in defining unconditionally secure group signature too. The concept of group signature in the unconditionally secure setting is new. We have introduced it in Chapter 5. With the introduction of group signatures in this contest, a new problem arises: who has the power to open a signature? This can be done by a Trusted Authority or with the help of the signer. In the first case, the TA distributes the secret information and opens group signature if required, while in the second case, the signer gives to the members of the group the proof that he has really generated the signature. For this reason two models for unconditionally secure group signature have been defined: one where the signature is opened by TA and the other where it is opened by the signer. For both models two constructions have been given: a symmetric construction and an asymmetric construction. In the symmetric construction a member of the group uses the same secret information both for signing and for verifying, while in
the asymmetric construction the secret information for signing and that for verifying are different.

Several open problems deserve further research. The most interesting and difficult one consists in finding a method to handle long messages. Indeed, current schemes require a message to be of a certain fixed length. As well as, better constructions aiming at minimizing user memory storage requirements and the size of the signatures, are topics of theoretical and practical relevance.
References


References


96 References


References


Appendix A

Proof of Theorem 2.4

**Theorem A.1** Let $P_I[i, L]$ and $P_S[i, L]$ be defined as in equation (2.1) and (2.2). Assume that $M \neq M'$, then

1. $P_I[i, L] \geq 2^{-I(M; E_i|E_L)}$.
2. $P_S[i, L] \geq 2^{-I(M'; E_i|M, E_L)}$.

**Proof:**

1. We define an impersonation characteristic function $X_I$ on $M \times E_i \times E_L$ by

$$X_I(m, e_i, e_L) = \begin{cases} 1 & \text{if } m \text{ is valid for } e \in E \text{ in } C \text{ such that } \tau_i(e) = e_i \text{ and } \tau_L(e) = e_L \\ 0 & \text{otherwise.} \end{cases}$$

From the definition of the impersonation attack we can express $P_I[i, L]$ as

$$P_I[i, L] = \max_{m \in M} P(\pi_i(m) \text{ is valid in } C_i|e_L \in E_L) = \max_{m \in M} \sum_{e_i \in E_i} X_I(m, e_i, e_L)P(e_i|e_L).$$

For a given $L \subseteq \{1, \ldots, n\}$ and $i \notin L$, let $P(m, e_i, e_L)$ be the joint probability distribution induced by the system. If $X_I(m, e_i, e_L) = 0$ then $P(m, e_i, e_L) = 0$. Indeed, if $P(m, e_i, e_L) \neq 0$ then $m$ is a valid message for $e$ with $\tau_i(e) = e_i$ and $\tau_L(e) = e_L$, which contradicts the definition of $X_I(m, e_i, e_L)$.

$$I(M; E_i|E_L) = E_{P(m, e_i, e_L)} \frac{P(m, e_i, e_L)}{P(M, E_i|M, E_L)P(e_i|E_L)}$$
Appendix A. Proof of Theorem 2.4

For each pair \((m, e_L)\) with \(P(m, e_L) \neq 0\), if \(X_I(m, e_i, e_L) = 0\) then \(P(e_i|m, e_L) = 0\). In this case, \(P(e_i|m, e_L)\log \frac{P(e_i|m, e_L)}{P(e_i|e_L)} = 0\). It follows that the summation taking over \(E_i\) in the above identity is restricted to all \(e_i\) for which \(X_I(m, e_i, e_L) = 1\). Thus we have

\[
I(M; E_i|E_L) = \sum_{m \in M, e_L \in E_L, P(m,e_L) \neq 0} P(m, e_L) \left( \sum_{e_i \in E_i} P(e_i|m, e_L)X_I(m, e_i, e_L) \log \frac{P(e_i|m, e_L)}{P(e_i|e_L)}X_I(m, e_i, e_L) \right).
\]

By applying the log-sum inequality we have

\[
I(M; E_i|E_L) \geq \sum_{m \in M, e_L \in E_L, P(m,e_L) \neq 0} P(m, e_L) \left( \sum_{e_i \in E_i} P(e_i|m, e_L)X_I(m, e_i, e_L) \right) \log \frac{\sum_{e_i \in E_i} P(e_i|m, e_L)X_I(m, e_i, e_L)}{\sum_{e_i \in E_i} P(e_i|e_L)X_I(m, e_i, e_L)}.
\]

For each pair \((m, e_L)\), as we have noted before, if \(P(m, e_L) \neq 0\) and \(X_I(m, e_i, e_L) = 0\), then \(P(e_i|m, e_L) = 0\). It follows that

\[
\sum_{e_i \in E_i} P(e_i|m, e_L)X_I(m, e_i, e_L) = 1
\]

and

\[
\sum_{e_i \in E_i} P(e_i|e_L)X_I(m, e_i, e_L) = P(\pi_i(m) \text{ is valid in } C_i|e_L).
\]

We obtain

\[
I(M; E_i|E_L)
\]
\[ \geq - \sum_{m \in M, e \in E_L} P(m, e_L) \log P(\pi_i(m) \text{ is valid in } C_i|e_L) \]

\[ = - \sum_{e_L \in E_L} P(e_L) \sum_{m \in M} P(m|e_L) \log P(\pi_i(m) \text{ is valid in } C_i|e_L). \]

Since

\[ P_{I}[i, L] \geq \sum_{e_L \in E_L} P(e_L) \left[ \max_{m \in M} P(\pi_i(m) \text{ is valid in } C_i|e_L) \right] \]

\[ \geq \sum_{e_L \in E_L} P(e_L) \left[ \sum_{m \in M} P(m|e_L) P(\pi_i(m) \text{ is valid in } C_i|e_L) \right], \]

by applying the Jensen inequality it follows

\[ \log P_{I}[i, L] \geq \sum_{e_L \in E_L} P(e_L) \sum_{m \in M} P(m|e_L) \log P(\pi_i(m) \text{ is valid in } C_i|e_L) \]

\[ \geq -I(M; E_i|E_L). \]

therefore, \( P_{I}[i, L] \geq 2^{-I(M; E_i|E_L)}. \)

2. In the substitution attack \( R_L \), receives their keys from the sender, observe a message \( m \) that is transmitted by \( T \) and substitutes another message \( m' \) for \( m \). \( R_L \) succeed if \( m' \) is accepted by \( R_i \) as authentic. We denote by \( P_{S}[i, L] \) the successful probability that \( R_L \) performs a substitution attack on \( R_i \). We have that

\[ P_{S}[i, L] = \max_{e_L \in E_L} \max_{m \in M} \max_{m' \in M, m' \neq m} P(\pi_i(m) \text{ valid in } C_i|m, e_L). \]

Now we define a substitution characteristic function \( X_S(m', m, e_i, e_L) \) by

\[ X_S(m', m, e_i, e_L) = \begin{cases} 1 & X_I(m', e_i, e_L) = 1 \text{ and } X_I(m, e_i, e_L) = 1, m' \neq m, \\ 0 & \text{otherwise.} \end{cases} \]

We introduce a random variable \( M' \) which only takes values when \( X_S(m', m, e_i, e_L) = 1 \). It follows that there is a joint probability distribution \( P(m', m, e_i, e_L) \) such that
Appendix A. Proof of Theorem 2.4

\[ P(m, e_i, e_L) \] is the probability distribution given in the system and such that if
\[ X_S(m', m, e_i, e_L) = 0 \] and \( P(m, e_i, e_L) \neq 0 \) then \( P(m', e_i, e_L) = 0 \).

\[
I(M'; E_i | M, E_L)
= \sum_{m' \in M', m \in M, e_i \in E_i, e_L \in E_L} P(m', m, e_i, e_L) \log \frac{P(m', m, e_i, e_L)}{P(m', m, e_i)}
\]

Then the summation over \( E \) follows that

\[
\text{Thus the summation over } E \text{ in the above identity is restricted to all } e_i \text{ for which } X_S(m', m, e_i, e_L) = 1. \text{ By applying the log-sum inequality, we have that}
\]

\[
I(M'; E_i | M, E_L)
= \sum_{m' \in M, m \in M, e_i \in E_i, e_L \in E_L} P(m', m, e_i, e_L) \log \frac{P(m', m, e_i, e_L)}{P(m', m, e_i)} X_S(m', m, e_i, e_L)
\]

Again, if \( P(m', m, e_L) \neq 0 \) and \( X_S(m', m, e_i, e_L) = 0 \) then \( P(e_i | m', m, e_L) = 0 \). It follows that

\[
\sum_{e_i \in E_i} P(e_i | m', m, e_L) X_S(m', m, e_i, e_L) = 0
\]
and

\[
\sum_{e_i \in E_i} P(e_i | m, e_L) X_S(m', m, e_i, e_L) = P(\pi_i(m')) \text{ is valid in } C_i | m, e_L)
\]
So we have

\[
I(M'; E_i|M, E_L) \\
\geq - \sum_{m' \in M', m \in M, e_L \in E_L} P(m', m, e_L) \log P(\pi_i(m')) \text{ is valid in } C_i|m, e_L \\
= - \sum_{m \in M, e_L \in E_L} P(m, e_L) \sum_{m' \in M'} P(m'|e_L, m) \log P(\pi_i(m')) \text{ is valid in } C_i|m, e_L
\]

Since

\[
P_S[i, L] \\
\geq \sum_{e_L \in E_L} P(e_L) \sum_{m \in M} P(m|e_L) \sum_{m' \in M'} P(m'|m, e_L) P(\pi_i(m')) \text{ is valid in } C_i|m, e_L \\
\geq \sum_{e_L \in E_L, m \in M} P(e_L, m) \sum_{m' \in M'} P(m'|m, e_L) P(\pi_i(m')) \text{ is valid in } C_i|m, e_L
\]

By applying the Jensen’s inequality, it follows

\[
\log P_S[i, L] \\
\geq \sum_{e_L, m \in E_L} P(e_L, m) \sum_{m' \in M'} P(m'|m, e_L) \log P(\pi_i(m')) \text{ is valid in } C_i|M, e_L \\
\geq -I(M'; E_i|M, E_L).
\]

We obtain

\[
P_S[i, L] \geq 2^{-I(M'; E_i|M, E_L)}.
\]