

THE FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS AND SCIENCES

ANONYMOUS T -OUT-OF- N THRESHOLD SIGNATURE SCHEMES

By

KAROLINA MANEVA-JAKIMOSKA

A Thesis submitted to the
Department of Computer Science
in partial fulfillment of the
requirements for the degree of
Master of Science

Degree Awarded:
Spring Semester, 2006

The members of the Committee approve the Thesis of Karolina Maneva-Jakimoska defended on April 3, 2006.

Mike Burmester
Professor Directing Thesis

Alec Yasinsac
Committee Member

Breno de Medeiros
Committee Member

The Office of Graduate Studies has verified and approved the above named committee members.

To my late father, who considered knowledge and education among the most important and valuable things. Wish you were here.

ACKNOWLEDGEMENTS

I would like to thank my family, for their unconditional support and love. I thank my mother Elica for her belief in me, my husband Goce for his endless understanding and patience, my son Nikola for his motivational energy and my sister and her family for their constant encouragement.

I would like to thank my major professor Dr.Mike Burmester for his invaluable guidance towards achieving my master's degree and for all the time that he dedicated to advise me on my research work during my studies. I thank the other members of my committee Dr.Alec Yasinsac and Dr.Breno DeMedeiros and I thank Tri Van Le. I appreciate their assistance and comments on my thesis.

I would also like to thank all the faculty and staff members from the Computer Science Department who dedicated part of their time to help me and support me throughout my graduate studies at Florida State University.

— Karolina

TABLE OF CONTENTS

List of Figures	vi
Abstract	vii
1. Introduction	1
2. Preliminaries	5
2.1 Secure ring signature schemes	5
2.2 Ad-hoc groups	7
2.3 Covering designs	8
3. Ring signatures	9
3.1 The Rivest-Shamir-Tauman ring signatures	9
3.2 Threshold Ring Signatures	12
4. New efficient threshold ring signatures based on covering designs	18
4.1 Formal definition and construction example	19
4.2 Security of the scheme	21
5. Ring signatures based on Vandermonde matrices	24
5.1 Security of the Vandermonde scheme	27
5.2 Extending the Vandermonde scheme to ID-based signatures	29
6. Beyond threshold ring signatures: General access structures	32
7. Efficiency	34
8. Conclusions	36
REFERENCES	37
BIOGRAPHICAL SKETCH	40

LIST OF FIGURES

3.1	Rivest-Shamir-Tauman ring signature scheme	10
3.2	The super-ring composition in the Bresson-Stern-Szydlo scheme with $t = 3$	14
4.1	A threshold ring signature scheme when $t = N/3$	19
4.2	d -cube ring set system construction for the cases: $d = 2$ and $d = 3$	21

ABSTRACT

In many multi-user cryptographic applications (e.g., electronic voting, digital lotteries, e-cash application, anonymous access to some resources, etc.), anonymity pops up as one of the main security objectives. Protecting private information about the involved users is not only desirable but crucial for existence and proper working of these applications.

Group signatures were the first signatures to provide anonymity of the signer(s): the members of the group can anonymously sign messages on behalf of the group using specially designed keys. The keys used by the individual members of the group are generated and distributed by a trusted group manager. Hence, group signatures are suitable for cooperative groups that have some preexisting structure. They are not suitable for groups that can be formed in an ad-hoc manner. To solve this problem, Rivest, Shamir and Tauman (ASIACRYPT 2001) introduced the notion of ring signature schemes. Unlike group signatures, ring signatures have no group managers. Any user can select a set (ring) of possible signers that includes himself, and using his private key and the public key of the other member of the ring, he can sign on behalf of the ring. Bresson, Stern and Szydlo (CRYPTO 2002) extend the notion to a threshold setting where some minimum number t of members of the group has to cooperate in order to sign a message. The complexity of the threshold ring signature scheme proposed by Bresson et al is prohibitively large even for relatively small sets of signers.

Our contribution: We propose two new anonymous signature schemes. The first one is a threshold ring signature scheme that is constructed using covering designs. This scheme is efficient even for large groups of signers. The cost we pay is that anonymity is not

perfect although it remains unconditional. The other one is a threshold scheme that is based on Vandermonde matrices. The second scheme is not always as efficient as the first one. However, it provides unconditional and perfect anonymity.

In a threshold ring signature scheme, any subset of members whose size is not below the threshold t can generate a signature. However, in some situations, we want to be able to specify which subsets can sign. We go beyond threshold ring signatures and propose a scheme where the possible subsets of actual signers are defined by a general access structure.

CHAPTER 1

Introduction

In the late eighties and early nineties, the Internet became not just a failsafe method of defense communications and a mean for researchers but a never ending resource for sharing data too. A lot of new applications appeared initiated primarily by e-commerce. These new applications demand new security requirements as well. Existence of a predefined groups and a way of their members to identify themselves as part of the group became inevitable. For many multi-user cryptographic applications such as electronic voting, digital lottery and e-cash applications, protecting private information was/is not only desirable but mandatory for cases where it might result in large financial loss. The integrity and non-repudiation provided by the existing digital signatures was not sufficient. Anonymity was essential in most cases.

In 1991, Chaum and Van Heyst [8] introduced the concept of group signatures. Many new requirements were identified and some improvements were proposed afterwards. However, the formalization and generalization of the definitions was just recently done [3, 4]. In group signature schemes, members are registered in groups administered by a group manager. Each member of a group can generate an anonymous signature on behalf of the group. Associated to the group is a single signature-verification key, also called a group public key. Each group member has its own secret signing key based on which it can produce a signature relative to the group public key. The group has its own manager, a trusted authority, who can trace any malicious activities and revoke the keys of all uncooperative members. The two main requirements for the group signatures remain the same, traceability and anonymity. Traceability is provided by the secret key that the manager holds. Using this key and a given signature he can extract the identity of the group member who created the signature. Anyone who does not hold this secret key should not be able to extract the identity of the actual

signer from a given group signature (i.e., the anonymity of the signer is guaranteed). The mechanism used to allow traceability, provides some level of security for non-signing members in case of dispute. Hence, group signatures are only appropriate tool when members have agreed to cooperate. However, for all groups that are formed “on the fly”, the group members don’t know each other and cooperation is questionable. The notion of group signatures is not suitable in this case. A concept that will address the requirements of the non-cooperative groups formed without previous setup is the one of ring signatures.

Ring signatures were introduced by Rivest, Shamir and Tauman [21]. The original motivation for their introduction was to provide a scheme that will allow anonymous leaking of secrets. Ring signatures can also be used to provide a member of a certain class of users with access to a particular resource without explicitly identifying this member. Their applicability to solve a plethora of problems has been already shown. Whenever a third-party verifiability is required, the ring signatures are more suitable than ad-hoc identification schemes [5, 12]. In the context of e-mail applications, ring signatures enable the sender to sign the e-mail with respect to the ring containing the sender and the receiver. In this case, the receiver knows who sent the message but can not prove it to any third party. To the outside world the e-mail can be signed by any of the two parties involved in e-mail correspondence.

The concept of ring signatures is related, but incomparable to the concept of group signatures. Ring signatures allow for greater flexibility: no centralized group manager or coordination among various users is required. Rings can be formed completely “on the fly” and in ad-hoc manner. The users have a choice over the level of anonymity they desire by generating any particular signature via selection of an appropriate ring. Any user can add his name/public key to an arbitrary set of other users he chooses and produce a ring signature revealing only that the anonymous author belongs to the set. This is infeasible with standard group signatures, where the possible signers by definition are registered members of the group. The absence of a revocation manager in the ring signatures allows for unconditional anonymity. In the case of group signatures, anyone that knows the secret key of the manager can determine the identity of the actual signer. Hence, unconditional anonymity cannot be achieved.

Another advantage of ring signatures over group signatures and some of the other solutions is their efficiency, or the possibility to be made very efficient. They exclude heavy

usage of asymmetric computations as zero-knowledge proofs or proofs of membership which makes them very fast. The efficiency gained in the ring signature schemes is also beneficial to cryptographic schemes built on top of ring signatures (e.g., multi-designated verifiers signatures, non-interactive deniable ring authentication and perfect concurrent signatures).

All these characteristics make ring signatures highly suitable for ad-hoc groups [12, 2, 19], which are computed instantly without involvement of third party and setup procedures. As many other groups, ad-hoc groups can also work in a threshold setting. In this case t out of N parties that belong to the ad-hoc group have to combine their knowledge to create certain signatures. Any t parties can perform the operation and create the signature. However, no $t - 1$ parties can succeed in doing so. These types of ad-hoc groups will be main point of interest when describing our new threshold schemes.

One of the shortcomings of ring signatures is their length. Their size grows linearly with the group size making them very impractical for large sets of signers. Dodis, Kiayias, Nicolosi and Shoup [12] introduced a new way of constructing ring signatures for anonymous identification in ad-hoc groups so that the size of the signature is constant. They have improved the efficiency of most of the ring signature schemes [21, 7, 14, 6]. However, they assume that the ring doesn't change much through the time, and that it is not necessary to include the public key of each member in the signature. We think that these assumptions restrict the dynamics in ad-hoc groups. The schemes that we propose are better suited for groups that change rapidly.

We propose two threshold ring signature schemes. The first scheme is based on covering designs, and it is more efficient than the schemes presented in [7, 14] especially for large set of signers. The scheme provides unconditional but not perfect anonymity. The second scheme utilizes Vandermonde matrices when constructing the signature. The computation of the signature is fast and easy to manage.

The rest of the thesis is organized as follows. In Section 2 we provide some definitions that are necessary to understand the concepts studied in the subsequent sections. In Section 3, we describe some ring signature schemes that are related to our new schemes. The description and security analysis of our new scheme based on covering designs is given in Section 4. Section 5 provides a description, security analysis and extension for ID based signatures of the t -out-of- N signature scheme based on Vandermonde matrices. In Section 6, we go beyond threshold ring signatures and propose use of general access structures to define the

possible subsets of actual signers. Section 7 compares our schemes with the existing proven secure schemes. The thesis ends with some conclusions.

CHAPTER 2

Preliminaries

This section provides definitions that will be used in the later sections. We follow the approach presented in [5].

2.1 Secure ring signature schemes

We first informally define ring signature scheme. A ring signature scheme consists of three algorithms:

Gen generates a public/secret key pair for each of the N users. The ordered list $\mathcal{R} = (P_1, \dots, P_N)$ of the public keys is called a public-key ring.

RingSign produces a ring signature σ on a message m , given m , ring of N users, their corresponding public keys $(P_1, \dots, P_s, \dots, P_N)$ and the secret key S_k of the actual signer.

RingVerify takes as an input a message m and a signature σ . It accepts the signature if σ is a valid signature on m , and outputs “true”. Otherwise, it rejects the signature and outputs “false”.

A basic requirement of any digital signature scheme is that signatures can not be forged. That is, it is computationally infeasible to produce a signature on a new message without the knowledge of the secret key S_k . A more formal definition based on the adaptive chosen message attack follows.

Definition 2.1 (Unforgeability) *A ring signature scheme $(\text{Gen}, \text{RingSign}, \text{RingVerify})$ is unforgeable if for any efficient adversary \mathcal{A} the probability that \mathcal{A} succeeds in the following game is negligible:*

1. Key pairs $\{(P_i, S_i)\}_{i=1}^{n(k)}$ are generated using $\mathbf{Gen}(1^k)$ and the set of public keys $R = \{P_i\}_{i=1}^{n(k)}$, $|R| = N$ is given to \mathcal{A} .
2. \mathcal{A} is given access to a signing oracle $\mathbf{OSign}(\cdot, \cdot)$ where $\mathbf{OSign}(s, m)$ outputs a ring signature $\mathbf{Sign}_{s, S_{k_s}}(m, R)$ on a message m , when the actual signer is s .
3. \mathcal{A} outputs (m^*, σ^*) and succeeds if $\mathbf{Verify}_R(m^*, \sigma^*) = \text{"true"}$ and it never queried (\star, m^*) to its signing oracle.

A threshold ring signature scheme consists of three algorithms:

\mathbf{Gen} generates a public/secret key pair for each of the N users.

$\mathbf{TRingSign}$ produces a (t, N) -ring signature σ on a message m , given m , a ring of N users with their corresponding public keys and the secret keys of t members. The number of signers t and the N public keys of the ring members are part of the signature σ .

$\mathbf{TRingVerify}$ takes as an input a message m and a threshold ring signature σ and outputs “true” accepting the signature if it is valid and false otherwise followed by rejection of the signature.

Same as ring signature schemes, threshold ring signature schemes are secure if they are anonymous and unforgeable. Unforgeability of threshold ring signatures schemes can be defined in a similar manner. The difference appears because threshold ring signatures allow the signers to choose an ad-hoc collection of users to contribute their share of the signature. In this case, there is a possibility that some of the users can be corrupted and cooperate with the adversary.

Definition 2.2 (t -out-of- N Unforgeability) *A t -out-of- N threshold ring signature scheme $(\mathbf{Gen}, \mathbf{TRingSign}, \mathbf{TRingVerify})$ is unforgeable if for any efficient adversary \mathcal{A} the probability that \mathcal{A} succeeds in the following game is negligible:*

1. Key pairs $\{(P_i, S_i)\}_{i=1}^{n(k)}$ are generated using $\mathbf{Gen}(1^k)$ and the set of public keys $R = \{P_i\}_{i=1}^{n(k)}$, $|R| = N$ is given to \mathcal{A} .
2. \mathcal{A} is given access to a signing oracle $\mathbf{OSign}(\cdot, \cdot)$ where $\mathbf{OSign}(T, m)$ outputs $\mathbf{Sign}_{T, S_T}(m, R)$ for a message m and set $|T| = t$ where $T \subseteq R$.

3. \mathcal{A} has access to a **CorruptOracle** and can successfully corrupt up to $(t - 1)$ secret keys S_i from T .
4. \mathcal{A} outputs (m^*, σ^*) and succeeds if $\mathbf{TRingVerify}_R(m^*, \sigma^*) = "true"$ and it never queried (\star, m^*) to its signing oracle

As we mentioned, unforgeability is not the only security requirement for the ring signature schemes. Anonymity that will ensure signer ambiguity is required also. The anonymity can be unconditional or computational based on the computational capability of the adversary. If the adversary has an unbounded computational power and still can not gain any information concerning the identity of the actual signer better than a random guess, then the anonymity provided by the signature scheme is unconditional. Computational anonymity on the other hand is provided when we assume that the adversary has limited computational power and can not find the identity of the signer better than a random guess. Our schemes provide unconditional anonymity. Bellow we provide more formal definition of unconditional anonymity.

Definition 2.3 (*t*-out-of-*N* Anonymity) *A t-out-of-N threshold ring signature scheme $(\mathbf{Gen}, \mathbf{TRingSign}, \mathbf{TRingVerify})$ is anonymous if for any efficient adversary \mathcal{A} the probability that \mathcal{A} succeeds in the following game is not significantly greater than a random guess:*

1. Key pairs $\{(P_i, S_i)\}_{i=1}^{n(k)}$ are generated using $\mathbf{Gen}(1^k)$ and the set of public keys $R = \{P_i\}_{i=1}^{n(k)}, |R| = N$ is given to \mathcal{A} .
2. \mathcal{A} is given access to a signing oracle $\mathbf{OSign}(T, \cdot)$, where $\mathbf{OSign}(T, m)$ outputs $\mathbf{Sign}_{T, S_T}(m, R)$ for a message m and a fixed secret t -set $T \subseteq R$.
3. \mathcal{A} outputs a t -set T' and succeeds if T' is equal to T .

2.2 Ad-hoc groups

We use the following definition of ad-hoc groups.

Definition 2.4 (Ad-hoc group) *An ad-hoc group Σ is a list of N users, including N certified public keys, accompanied by a list of subsets S_j of these users, called the acceptable subsets. This second list may be optionally replaced with a predicate defining exactly which subsets are acceptable.*

This means that ad-hoc signature will retain maximal anonymity, and at the same time, it will prove that the signing members all belong to at least one of these acceptable subsets.

2.3 Covering designs

The main definitions regarding covering designs that we use as a basis for our schemes are provided below.

Definition 2.5 (Set system) *A set system is a pair (X, \mathcal{B}) consisting of a set $X = \{a_1, a_2, \dots, a_v\}$ and a multiset \mathcal{B} whose elements are subsets (or blocks) of X .*

Definition 2.6 (Covering design) *The set system (X, \mathcal{B}) is a (v, b, t) -covering design, where v, b, t are integers and $t \leq b \leq v = |X|$, if*

1. *all blocks in \mathcal{B} are b -subsets of X , i.e. $\forall B \in \mathcal{B} |B| = b$, and*
2. *any t -subset of X is contained in at least one block.*

Definition 2.7 (Complementary set system) *The complement of a set system (X, \mathcal{B}) is the set system (X, \mathcal{B}^c) , where $\mathcal{B}^c = \{X \setminus B_i | B_i \in \mathcal{B}\}$.*

The last definition will be crucial in our proof of security for the scheme based on covering designs.

CHAPTER 3

Ring signatures

In this section, we provide detailed description of concepts that will be of significant importance for us to introduce the new schemes.

3.1 The Rivest-Shamir-Tauman ring signatures

When the concept of ring signatures was presented for the first time [21], it was said to be ideal for “leaking a secret”. In order to explain this “leaking”, let us imagine that there is a company AB which is involved in some illegal activity. One of the employees who also belongs to the Board of Directors does not agree with the company’s policies. Using a ring signature scheme, he can expose the company without revealing his identity. He will sign the message in the name of the Board of Directors, and then send it to a journalist. Based on the ring signature, the journalist will be able to check that the message is correct and comes from a member of the Board of Directors. At the same time, the journalist will not have a clue who is the actual composer and signer of the message.

As we already mentioned in the Introduction, with ring signatures the group formation can happen in ad-hoc fashion: no manager is involved and the signer can include anyone in his group. The actual signer chooses a group of individuals and uses their public keys together with his secret key to compute the signature. Each time the signer wants to sign a message, he can choose different set of public keys to be part of the construction of the signature. The non-signers might not even know that they have been included in a certain group. This is a sharp contrast comparing to the group signature setting where all of the users execute a Join protocol with the group manager to obtain a group membership certificate. Ring signatures provide complete anonymity for the signer. When the verifier verifies the signature the only thing that he knows about the actual signer is that he is the part of the group which signed

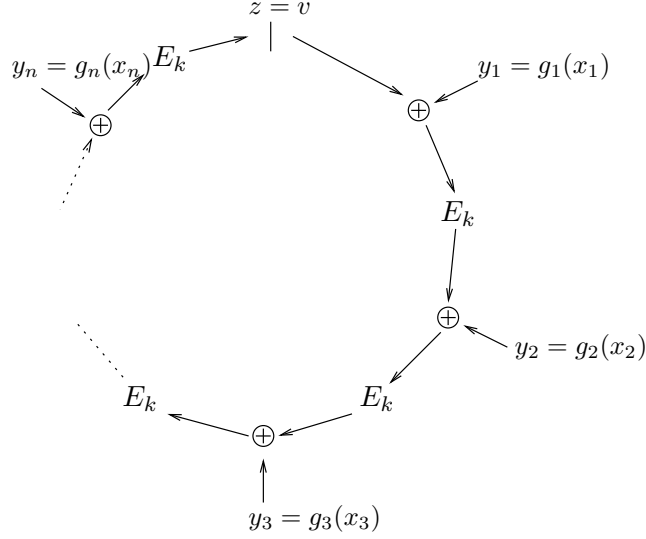


Figure 3.1: Rivest-Shamir-Tauman ring signature scheme

the message. The true identity of the signer can not be revealed.

The RSA version of the ring signature scheme presented in [21] uses one-way trap-door permutations that have a common domain. These one-way trap-door permutations are constructed by modifying the original RSA one-way permutations as follows. All public keys $P_i(n_i, e_i)$ have different n_i 's and the results $f_i(x) = x^{e_i} \pmod{n_i}$ from the standard RSA one-way permutation have different domain sizes. Their domains can be unified by choosing $\{0, 1\}^b$ as their domain where 2^b is larger than any of the n_i 's and b is the block size of the message. First for any b -bit input m two non-negative integers q_i and r_i has to be defined such that $m = q_i n_i + r_i$ and $0 \leq r_i \leq n_i$. Next for each trap-door permutation f_i over Z_{n_i} the extended trap-door permutation g_i over $\{0, 1\}^b$ is defined as:

$$g_i(m) = \begin{cases} q_i n_i + f_i(r_i) & \text{if } (q_i + 1)n_i \leq 2^b \\ m & \text{else} \end{cases} \quad (3.1)$$

As it can be seen from the formula, the extended one-way function is changing the lower bits of m and the rest remains unchanged. The input m will remain completely unchanged only if the result happens to be larger than $2^b - 1$. One way to ensure that this will never happen is by choosing b to be at least 160 bits longer than any of the n_i 's. Fig. 3.1 depicts the Rivest-Shamir-Tauman ring signature scheme [21]. The s -th member of a ring with size N , will produce a signature σ on a message m as follows:

1. Let E_k be a block encryption function with key length l and block size b .
2. Compute a key k as $k = h(m)$, where $h(\cdot)$ is a publicly known collision resistant function that maps arbitrary inputs to strings of length l .
3. Pick a random glue value $v \in \{0, 1\}^b$.
4. Pick random $x_i \in \{0, 1\}^b$ values, for $1 \leq i \leq N$ and $i \neq s$ and compute $y_i = g_i(x_i)$ where each g_i is extended trap-door permutation.
5. Solve for y_s the ring equation:

$$E_k(y_N \oplus E_k(y_{N-1} \oplus E_k(\dots E_k(y_1 \oplus v)))) = v. \quad (3.2)$$

6. Use the trap-door s_s to invert g_s on y_s and obtain the value of $x_s = g_s^{-1}(y_s)$.
7. Output the ring signature σ as a $(2N + 1)$ -tuple

$$\sigma = (P_1, P_2, \dots, P_N; v; x_1, x_2, \dots, x_N). \quad (3.3)$$

When the verifier receives a message m with a signature σ he verifies that someone from the ring has signed the message. The provided information in the signature itself doesn't reveal the identity of the actual signer. The verifier computes $y_i = g_i(x_i)$, for $1 \leq i \leq N$, obtains the key $k = h(m)$, and checks the ring equation:

$$E_k(y_N \oplus E_k(y_{N-1} \oplus E_k(\dots E_k(y_1 \oplus v)))) = v. \quad (3.4)$$

If the equation holds the verifier accepts the ring signature and outputs "true", and rejects the signature otherwise.

This scheme is proven to be secure. It is unforgeable and provides unconditional anonymity for the signer in the sense that even a computationally unbounded adversary that has access to infinitely many chosen-message ring signatures cannot guess the identity of the actual signer with any advantage. The unforgeability of the ring signature is based on the hardness of inverting the extended RSA permutation g_i . This is an equivalent to the RSA assumption because only someone who knows how to invert f_i can invert g_i with more than a negligible probability. For more details on the proof see [21].

3.2 Threshold Ring Signatures

Threshold ring signatures prove that certain minimum number of members of a group must have collaborated to produce the signature, while hiding the precise membership of the subgroup of signers. In particular, no group of less than t members can forge the signature in collaboration with the adversary, and the verifier cannot determine the identity of the actual signers.

3.2.1 The Bresson-Stern-Szydlo signatures

Based on the ring signature scheme described in the previous section, Bresson, Stern and Szydlo(BSS) [7] provided extension suitable for threshold schemes and ad-hoc groups. The ring of N users is partitioned into t disjoint subsets a number of times. Each partition is one node in a super-ring and each subset of the partition is one sub-ring. A group of t users can close the super-ring (i.e., produce a (t, N) -ring signature) if and only if one of the partitions in the super-ring is a fair partition for the group of t users. Each subset of the fair partition contains at least one actual signer.

As a part of their scheme, Bresson, Stern and Szydlo proposed a use of a perfect hash functions [1] instead of the symmetric encryption function $E_{h(m)}(x)$. This perfect hash function is used to construct the partitions of the super-ring. BSS scheme modifies the underlying ring signature scheme so that the signer can start the calculations from his own position (the previous scheme has to start the calculations from the node number 1). The actual signer chooses a random seed ω and computes along the ring:

$$\begin{aligned}
 v_{i_s+1} &= h(m, \omega) \\
 v_{i_s+2} &= h(m, v_{i_s+1} \oplus g_{i_s+1}(x_{i_s+1})) \\
 &\vdots \\
 v_{i_s} &= h(m, v_{i_s-1} \oplus g_{i_s-1}(x_{i_s-1}))
 \end{aligned} \tag{3.5}$$

Before the signer closes the ring, he computes the last input x_{i_s} by using his own secret key to solve the equation $v_{i_s} \oplus g_{i_s}(x_{i_s}) = \omega$. The signer produces the signature σ as:

$$\sigma = (P_1, P_2, \dots, P_N; i_0; v_{i_0}; x_1, x_2, \dots, x_N) \tag{3.6}$$

by choosing at random the position i_0 in the ring and providing the corresponding v_{i_0} value for that position.

Furthermore, the ring does not have to close perfectly. A gap γ can be allowed between any two indexes in the ring as long as the value of i_γ (the position of the gap), γ (the value of the gap) and v_{i_γ} (the glue value for the gap) are included in the signature. During the verification process, the verifier has to calculate $v_{i_\gamma} \oplus \gamma$ at the place of the gap. In this case, even if there is no signer in the ring we can simulate it by choosing γ and making the final result to be ring signature. However, the security of the scheme is not intact. Usage of the gap is only acceptable if we assume that the adversary is not in the position to choose γ , otherwise the ring can be simulated and the scheme will be broken.

Below, a more detailed description of the BSS (t, N) ring signature scheme is provided. For simplicity, without loss of generality one can assume that the gap is between the first and the last node in the calculation ring (i.e., $i_0 = i_\gamma = 1$) and omit i_γ from the signature. The length of the signature will not be shortened because of this. The value of t has to be included in the signature. Only t members can construct the signature and no $t - 1$ members are enough to sign the message.

This scheme can be made more applicable to ad-hoc groups if the signature includes legitimate subsets. The reason for this is the very nature of the ad-hoc groups where the members of the group can belong to different subsets. However, not all of these subsets can produce a legitimate signature based on the standing that particular nodes have in the group. So the first step in the verifying process checks if all of the signing members belong to at least one acceptable subset. In this case, corrupting t members is not enough to forge a signature, they also have to belong to an acceptable subset. The acceptable subsets structure is specified using fair partitions. The group of ring members is divided into disjoint subgroups. The number of subgroups will be the same as the threshold value because the goal is to have at least one signer in each subgroup. However, if the set is partitioned only once and in each subgroup there is only one signer the anonymity of the actual signers is in jeopardy. Each signer is not anonymous over set of N possible signers but over the subset to which he belongs. Because of this the group is split many times. There should be at least one split with exactly one signer in each subgroup.

Let $\pi = (\pi^1, \pi^2, \dots, \pi^t)$ be a partition over $[1, N]$ in t subsets and $I = \{i_1, i_2, \dots, i_t\}$ be a set of indexes $\in [1, N]$. If all indexes belong to different subsets, then π is a fair partition with respect to I . If for any set of cardinality t there exists a fair partition in Π for I , we say that Π is (N, t) complete partitioning system. As we mentioned earlier, this

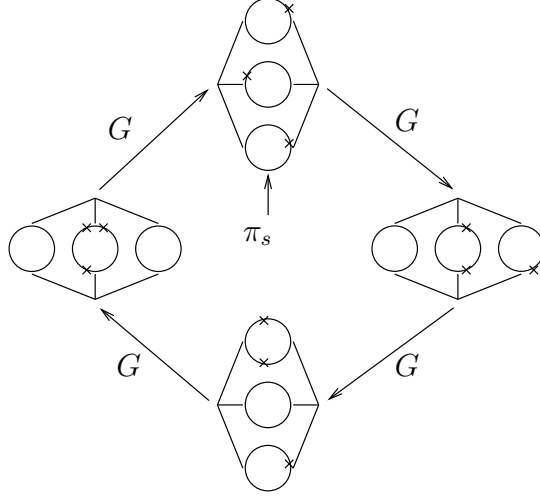


Figure 3.2: The super-ring composition in the Bresson-Stern-Szydlo scheme with $t = 3$.

can be achieved using perfect hash functions. It was shown in [1] that the size of (N, t) family of perfect hash functions is $p = 2^t \log(N)$ and the complete partitioning system will consist of p partitions: $\pi = \{\pi_1, \pi_2, \dots, \pi_p\}$. Fig. 3.2 depicts the basic principle behind the threshold ring signature scheme when $t = 3$. In this case π_s is a fair partition with respect to the three signers. Let $P_{i_1}, P_{i_2}, \dots, P_{i_t}$ be the t users that collaborate to sign a message m . Furthermore, assume that π_s is a fair partition for $I = \{i_1, i_2, \dots, i_t\}$ (as given in Fig. 3.2) and assume that for each $j \in [1, t]$ there is $i_j \in \pi_s^j$. The steps of the signing algorithm can be formalized through 5 subroutines:

1. During Choose Seeds the signers choose random seeds for each sub-ring in each partition.

```

Choose Seeds{
  for  $i = 1, \dots, p$ 
    for  $k = 1, \dots, t$ 
      choose random values  $v_i^k \in \{0, 1\}^b$ 
}

```

2. For each partition that does not have at least one signer in each subring, a ring-like

mechanism has to be simulated using the Ring Simulation subroutine:

```

Ring Simulation{
  for  $i = 1, \dots, p$  and  $i \neq s$ 
    for  $j = 1, \dots, N$ {
      choose random  $x_i^j \in \{0, 1\}^b$ 
      calculate  $y_i^j = g_i(x_i^j)$ 
      for  $k = 1, \dots, t$  and  $j \in \pi_i^k$ {
        calculate  $z_i^k = h(m, y_{i-1}^j \oplus$ 
           $h(m, y_{i-2}^j \oplus \dots \oplus h(m, \gamma \oplus y_\gamma^j \oplus h(\dots h(m, y_1^j \oplus v_i^k) \dots))) \dots)$ 
        calculate  $\gamma_i^k = v_i^k \oplus z_i^k$ 
      }
    }
}

```

- For each node in the super ring, a gap has to be calculated and the resulting input and output for the fair partition has to be obtained. First we choose random seed $\sigma_s \in \{0, 1\}^{bt}$ for the next node from the fair partition in the super-ring and calculate $u_s = G(\sigma_s)$. Here $G()$ is just a random hash function that returns $(t \times b)$ strings.

```

Super Ring{
  randomly choose  $\sigma_s$  from  $\{0, 1\}^{bt}$ 
  calculate  $u_{s+1} = G(\sigma_s)$ 
  for  $i = s + 2, \dots, p, 1, \dots, s$ 
    compute  $u_i = G(u_{i-1} \oplus (\gamma_{i-1}^1 || \dots || \gamma_{i-1}^t))$ 
  }

```

- Next, by closing the super-ring compute the gap values for the sub-rings of π_s , $(\gamma_s^1 || \dots || \gamma_s^t) \leftarrow u_s \oplus \sigma_s$. This result has to be parsed in order to get the “gaps” for each sub-ring in the fair partition. With that the ring will be closed and the sub-rings in the partition π_s will be solved. For all non-signers in the sub-ring, we execute Ring Closure 1, and for the signers, we execute Ring Closure 2:

```

Ring Closure 1 {
  for  $j \in [1, 0] \setminus I$ 
    choose random  $x_i^j$  over  $\{0, 1\}^b$ 
    compute  $y_s^j \leftarrow g_j(x_s^j)$ 
  }

```

```

Ring Closure 2 {
  for  $j \in I$  and  $j \in \pi_s^k$ 
    choose random  $\sigma_k$  from  $\{0, 1\}^b$ 
    compute  $y_s^j = C_{j, \gamma_s^k, m}^{-1}(\sigma_k, y_s^j, j \in \pi_s^k(R))$ 
    compute  $x_s^j = g_j^{-1}(y_s^j)$ 
  }

```

In the Ring Closure 2 subroutine the function C is just a short notation for the calculations done on each sub-ring. The computed signature σ at the end of the procedure is

$$\sigma = (v, u_v, \bigcup_{1 \leq i \leq p} (x_i^1, \dots, x_i^N, v_i^1, \dots, v_i^t)),$$

where v is randomly chosen index over $[1, p]$ that indicates to the verifier the starting point of the calculations. During the verification process the verifier first solves all the rings and then verifies the super-ring starting from position v .

As we can see from the final construction of the signature, the signature size grows with the number of members whose shares are part of the signature. That makes the scheme very inefficient when t is as large as N . In the next section, we will shortly describe a scheme suitable and more efficient than BSS scheme when t is as large as N .

3.2.2 An $(N - t)$ -out-of- N threshold ring signature scheme

An $(N - t)$ -out-of- N threshold ring signature scheme was recently proposed by Isshiki and Tanaka [14]. It is based on the schemes described in the previous two sections but is more efficient when the threshold $N - t$ is of the same order as N . Instead of super-ring the simple ring structure is kept but the trap-door one-way permutation is changed. While in BSS fair partitions are used to obtain correctness and anonymity, in this scheme, fair partitions provide for correctness and unforgeability. Since t in this scheme stands for non-signers if we just partition the set $[1, N]$ in t subgroups we can ensure that in every sub-group we have one non-signer. The final goal is to have at least one subgroup where everybody is a signer. This subgroup is called legal subgroup. To preserve unforgeability, instead of making t partitions, $t + 1$ partitions are constructed. That way any $t + 1$ users are in the different subgroups in the partitioning system. The complete partitioning system will be $\Pi_N^{t+1} = \{\pi_1, \dots, \pi_p\}$ and each partition is $\pi_i = (\pi_i^1, \dots, \pi_i^{t+1})$.

Each partition subgroup has different number of elements. In the description of the scheme given below, by π_i^j we denote any partition from the complete partitioning system and by q_i^j the number of elements in it. The representation of the partitioning system will be $\pi_i^j = \{p_i^{j,1}, \dots, p_i^{j,q_i^j}\}$. S_i^j is used to denote the enlarged partition and Q is the maximum value of q_i^j . The scheme works as follows:

1. First step of the signing algorithm in this scheme is to make all of the subgroups to have equal number of elements by repeating the value of the last element. So, for each

partition where $q_i^j = Q$, $S_i^j = \pi_i^j$ else $S_i^j = \{\pi_i^j \cup \{p_i^{j,q_i^j+1}, p_i^{j,q_i^j+2}, \dots, p_i^{j,Q}\}\}$. In the second case, all the elements after the last one are repetition of that value until the number of elements reaches Q : $p_i^{j,q_i^j+1} = p_i^{j,q_i^j+2} = \dots = p_i^{j,Q} = p_i^{j,q_i^j}$

2. For each subgroup define a trap-door one way permutation G_i^j as:

$$G_i^j(x_1, x_2, \dots, x_Q) = g_{p_i^{j,1}}(x_1) \parallel \dots \parallel g_{p_i^{j,Q}}(x_Q)$$

where $g_{p_i^{j,k}}$ for $k = 1, 2, \dots, Q$, are extended one-way trap-door permutation on $P_{p_i^{j,k}}$ over $\{0, 1\}^b$. Also the assumption is that in each partition the subgroup S_i^j is a legal subgroup.

3. The signers choose random seeds s^1, s^2, \dots, s^Q from $\{0, 1\}^b$ and compute: $v_{j+1} = h(m, s^1, s^2, \dots, s^Q)$.
4. For each $k = j + 1, \dots, t + 1, 1, 2, \dots, j - 1$ the signers choose at random from $\{0, 1\}^b$, $x_k^1, x_k^2, \dots, x_k^Q$ and compute: $v_{k+1} = h(m, v_k \oplus G_i^k(x_k^1, \dots, x_k^Q))$.
5. Using his knowledge of the trap-door of $g_{p_i^{j,k}}$, each signer inverts G_i^j and obtains $x_j^1, x_j^2, \dots, x_j^Q$ such that $v_{j+1} = h(m, v_j \oplus G_i^j(x_j^1, x_j^2, \dots, x_j^Q))$.
6. From each partition signers output their piece of the signature as $(2(t+1)Q+2)$ -tuple:

$$\sigma_i = (P_{p_i^{1,1}}, \dots, P_{p_i^{1,Q}}, P_{p_i^{2,1}}, \dots, P_{p_i^{t+1,Q}}; i_0; v_{i_0}; x_1^1, \dots, x_1^Q, x_2^1, \dots, x_{t+1}^Q) \quad (3.7)$$

The whole signature will be a p -tuple of the signatures given above:

$$\sigma = (\sigma_1, \dots, \sigma_p)$$

The verifier verifies the signature by checking each σ_i on the message m . For each $k = i_0 + 1, i_0 + 2, \dots, t + 1, 1, 2, \dots, i_0 - 1$, the verifier computes:

$$v_k = h(m, v_{k-1} \oplus Q_i^k(x_k^1, \dots, x_k^Q)) \quad (3.8)$$

end checks the equation:

$$v_{i_0} = h(m, v_{i_0-1} \oplus G_i^j(x_{i_0-1}^1, \dots, x_{i_0-1}^Q)) \quad (3.9)$$

If for all σ_i the equations are satisfied, the verifier accepts the signature and outputs “true”. Otherwise, it rejects the signature outputting “false”.

CHAPTER 4

New efficient threshold ring signatures based on covering designs

In this section of the thesis, we will present a new threshold scheme which is based on covering designs and presented in [16]. The idea of using covering designs with threshold schemes as well as to achieve anonymity is not so new (e.g.[11, 20]), and there is a lot of work done in the area of efficient construction of covering designs (e.g. [18, 20]). The new scheme defined in this section utilizes the efficient constructions of covering designs for producing a threshold like ring signature. This scheme is suitable for ad-hoc groups where acceptable subsets of nodes who can produce valid signatures has to be defined in advance. These subsets are created based on the standing level that node has in the group. As an underlying scheme for our new scheme we can use any proven secure ring signature scheme. To be proven secure, the underlying scheme has to be unconditionally anonymous and computationally unforgeable. Let first briefly describe the idea behind our scheme. Consider a group of N possible signers $U = \{u_1, \dots, u_N\}$ and among them a set of actual signers $U_s = \{u_{s_1}, \dots, u_{s_t}\}$. The actual signers construct a collection of rings \mathcal{R} . Each ring \mathcal{R}_i in \mathcal{R} is an r -subset of U and contains at least one actual signer $u_{s_i} \in U_s$ and $1 \leq i \leq t$. We refer to the set system (U, \mathcal{R}) as the *ring set system* of the scheme. Whenever the users in U_s want to anonymously leak a secret m , they use a ring signature scheme to generate a ring signature σ_i on m for each ring $\mathcal{R}_i \in \mathcal{R}$. The threshold ring signature σ will be a p -tuple of the computed ring signatures: $\sigma = (\sigma_1, \dots, \sigma_p)$ where $1 \leq p \leq |\mathcal{R}|$. Since the sets that can sign the message are predefined, not any t -set of users can sign the message m . However the number of the sets that can sign m grows exponentially with N . A simple example with $t = N/3$ is depicted in Fig. 4.1. In this scheme, there are $N/3$ disjoint rings in \mathcal{R} and the number of t sets of users that can sign the message is $3^{N/3}$. Note that the complexity of the scheme is linear in the number of users

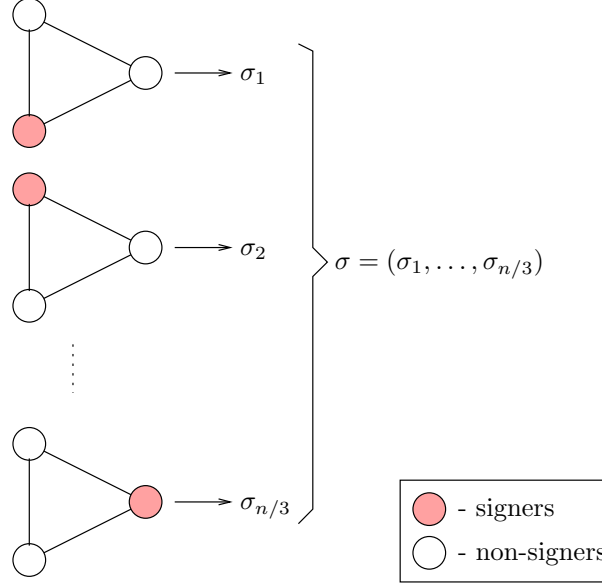


Figure 4.1: A threshold ring signature scheme when $t = N/3$

(i.e $\theta(N)$), while Bresson et al scheme in this case would have $\mathcal{O}(2^{N/3} N \log N)$ complexity.

4.1 Formal definition and construction example

In the previous section we gave a short description of our scheme based on covering designs. Now, we are going to give a formal description and a concrete construction example.

We referred to the group of possible signers as U such that $|U| = N$. Let U_s be the set of actual signers such that $U_s \subseteq U$ and $|U_s| = t$.

1. Each member (signer) $U_{s_i} \in U_s$ constructs one or more rings \mathcal{R}_i of size r . Each ring will have at least one signer (the actual constructor of the ring), which will ensure that the construction of the ring is feasible i.e the ring can be closed. Choosing additional signers as part of the ring is not necessary for the ring closure but considered more secure. As an underlying scheme for the ring construction any proven secure ring signature scheme can be chosen.
2. *Ring set system* \mathcal{R} is constructed from all the rings constructed by the signers. In this case, $r \leq p$ where $p = |\mathcal{R}|$. Construction of the *ring set system* can be done in many

different ways, however for a given set of signers this ring set is constructed in the beginning of the collaborated signing and it will be used by that set of signers until the actual set exists. A signer can belong to more than one *ring set system*. The signer participates in a construction of a *ring set system* for each group that he wants to be part of the signing process. However, we have to point out that if one signer belongs to multiple t -sets of signers, then the anonymity might be in jeopardy. We discuss the anonymity of the scheme only in the case when a single signer does not belong to multiple t -sets of signers.

3. The signing subset produces the signature by making a p -tuple of all the signatures produced by different rings from the same ring set system:

$$\sigma = \sigma_1, \sigma_2, \dots, \sigma_p \tag{4.1}$$

Each σ contains a list of public keys used in the construction of the ring. During the verification process the verifier verifies each σ_i separately. The signature is valid if all σ_i 's are valid. Even though the verifier have the information which ring set system signed the message the actual signer is still anonymous. However, the anonymity is not over all possible t -sets of users, but over the t -sets in the ring set systems.

In the example that we present here, the number of possible signers is $N = t^d$, where t is the number of actual signers and d is integer such that $d > 1$. Two different constructions of the ring set system in a d -cube for $d = 2$ and $d = 3$ are given on Fig. 4.2. All N participants are arranged in a d cube. We choose a covering design that will “cover” all the columns and all the rows of one side of the cube. Each user u_{i_1, \dots, i_d} , where $\{i_1, \dots, i_d\} \in \{0, 1, \dots, t - 1\}$ is a member of exactly d rings of the ring set system:

$$\begin{aligned} &\{u_{j_1, \dots, j_d} | j_1 = i_1\} \\ &\{u_{j_1, \dots, j_d} | j_2 = i_2\} \\ &\quad \vdots \\ &\{u_{j_1, \dots, j_d} | j_d = i_d\} \end{aligned}$$

Each user can be an actual signer. That is, for each user u there is a t -set that includes u and can produce a threshold ring signature. The number of t -sets that can generate signature is $(t!)^{d-1}$ but it will grow fast with both t and d . We will elaborate the efficiency

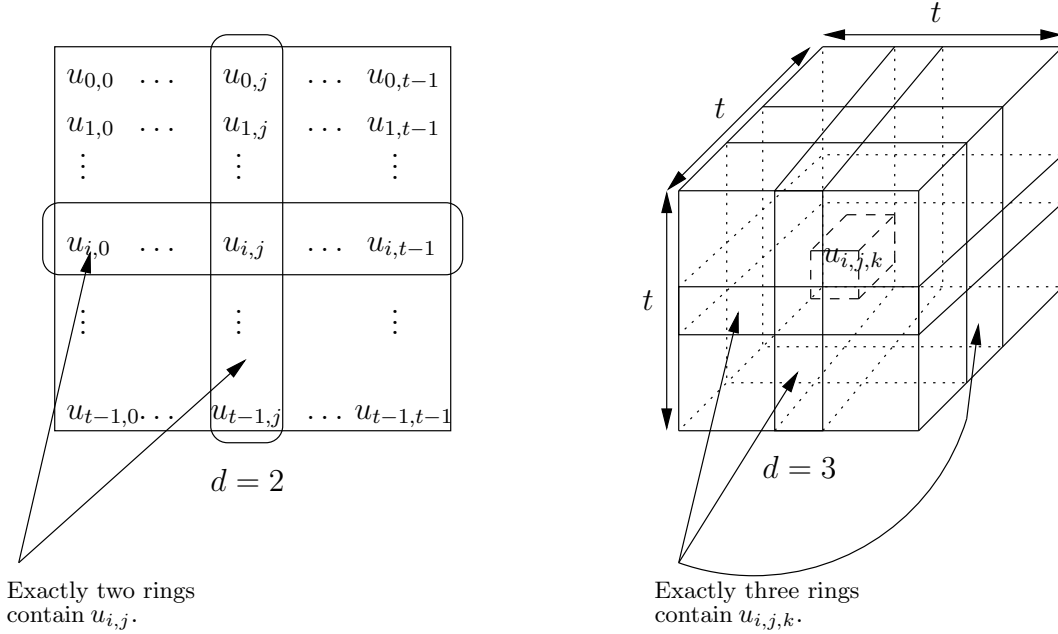


Figure 4.2: d -cube ring set system construction for the cases: $d = 2$ and $d = 3$.

of the scheme in one of the following sections but one can observe that the complexity of the scheme is $\mathcal{O}(dN) = \mathcal{O}(N \log N)$.

4.2 Security of the scheme

It is clear that t actual signers can generate a threshold ring signature since there is at least one signer in each ring $\mathcal{R}_i \in \mathcal{R}$. However, one must show that no $t - 1$ members can conspire and forge a signature.

We assume that the underlying ring signature scheme is unforgeable as in Definition 2.1. The adversary has knowledge of the public keys of r members of the ring. However, he does not know the corresponding secret keys. The adversary can query a signing oracle with a message m and a set of public keys r on what the oracle outputs a ring signature of m for the ring specified by the public keys. The signature produced by the oracle is valid only if the provided public keys belong to users in an actual signers subset. The scheme is unforgeable if the adversary can not produce a ring signature on a message that has not been signed previously.

In the case of a threshold ring signature scheme, we assume that the adversary has knowledge of the public keys of N users and the secret keys of $t - 1$ users. The adversary can send signing queries to an oracle to get threshold ring signatures. The goal of the adversary is to produce a threshold ring signature on a message m that was not previously sent for signing. The scheme is unforgeable if the adversary cannot succeed with non-negligible probability.

The following theorem proves that the threshold ring signatures described in the previous section are unforgeable and unconditionally anonymous.

Theorem 4.1 *Let the underlying ring signature scheme be unforgeable and unconditionally anonymous. Our threshold ring signature scheme (described in Section 4.1) is unforgeable and unconditionally anonymous if and only if the complement (U, \mathcal{R}^c) of the ring set system (U, \mathcal{R}) is an $(N, N - r, t - 1)$ -covering design.*

Proof: Suppose that the complement (U, \mathcal{R}^c) of the ring set system (U, \mathcal{R}) is not an $(N, N - r, t - 1)$ -covering design. This means that there is a $(t - 1)$ -subset $C \in U$ such that $C \cap \mathcal{R}_i \neq \emptyset$ for all $\mathcal{R}_i \in \mathcal{R}$. In other words, there is at least one conspirator (member of C) in each ring of the threshold scheme. Clearly, the $t - 1$ conspirators in C can forge signatures.

Assume now that the complementary set system (U, \mathcal{R}^c) of the ring set system (U, \mathcal{R}) is an $(N, N - r, t - 1)$ -covering design, but the threshold scheme is not unforgeable. We will show how a successful adversary for the threshold ring signature scheme can be converted into a successful adversary for the underlying ring signature scheme which is in contradiction to our assumption that the underlying ring signature scheme is unforgeable. The adversary for the ring signature scheme follows the same procedure as the adversary for the threshold scheme, and simulates the threshold scheme oracle. We can simulate the signing oracle for the threshold ring signature scheme using an oracle of a ring signature scheme as follows. Whenever the threshold scheme adversary sends a signing query, we construct a threshold ring signature by sending $|\mathcal{R}|$ signing queries to a ring signature scheme oracle, combining the answers and then sending the result back to the threshold scheme adversary. At the end, the threshold scheme adversary will output a threshold ring signature $\sigma^f = (\sigma_1^f, \dots, \sigma_{\mathcal{R}}^f)$ of a message that was not signed previously by the oracle. Suppose that the adversary knows the secret keys of the users $u_{i_1}, \dots, u_{i_{t-1}}$. Since (U, \mathcal{R}^c) is a $(N, N - r, t - 1)$ -covering, there is

a ring $\mathcal{R}_i \in \mathcal{R}$ such that $\mathcal{R}_i \cap \{u_{i_1}, \dots, u_{i_{t-1}}\}$ is empty. In other words, there is no user in \mathcal{R}_i whose secret key is known to the adversary. Therefore, the ring signature σ_i^f corresponding to the i -th ring \mathcal{R}_i is a forgery for the underlying ring signature scheme. \square

As we already mention, the ring set system is constructed only once (at the beginning) for a given group of actual signers, and the actual signers use this ring set system to generate all their signatures. Therefore, the anonymity of the scheme is unconditional. Even an adversary with unbounded computational power that has access to arbitrarily many chosen-message threshold ring signatures constructed using a given ring-set system cannot distinguish between the group of actual signers and any t -set that can generate threshold signatures.

The anonymity of the scheme depends on how the ring set systems are constructed. Different covering designs can be used for construction of the ring set system. However, different constructions can have different “levels” of anonymity. This means that the anonymity might not be perfect. Namely, although the number of t -sets of signers that can generate a threshold ring signature can be large, there are sets of t signers that can not generate valid signature for that particular ring set system. Whether or how much the membership of a single user in more than one t -set of signers can decrease the anonymity depends on how the ring set systems are constructed also. We leave this problem as an open question.

CHAPTER 5

Ring signatures based on Vandermonde matrices

In this section, we present new scheme for anonymous group signatures based on Vandermonde matrices. First, we will outline the framework for single-signer group signature ($t = 1$). Then, we will extend the scheme for any number of collaborating signers t such that $t \leq N$, where N is the number of members in the group. We use **RingSign** and **RingVerify** to denote the signing and verifying algorithms correspondingly. The signature produced and outputted by the **RingSign** algorithm has to be signer-ambiguous and unforgeable which will prevent the verifier of revealing the identity of the actual signer. As will be proven in the next section, this signature scheme satisfies the security conditions.

Let us briefly describe the scheme when there is only one signer. Consider a group of N members and one signer who wants to sign a message on behalf of the group. He chooses $N - 1$ random values and encrypts them using the public keys of his co-signers from the group. He constructs a linear polynomial using the computed values and the hash value of the message concatenated with some random value. The hash of the message is used instead of the actual message for improved security. From the polynomial, the signer computes his “random” value by inverting his one-way trap-door function. The signature contains all random values and the probability that the verifier will be able to reveal the identity of the signer is no better than $1/N + \epsilon$ where ϵ is negligibly small. Below we give a formal description of the scheme.

Let $U = \{U_1, \dots, U_N\}$ be the set of all possible signers. Each signer U_i has its own public key P_i and a corresponding secret key. We assume that all of the public keys are RSA keys. That means that every public key $P_i = (e_i, n_i)$ specifies one-way trap-door permutation on Z_{n_i} :

$$f_i(x) = x^{e_i} \pmod{n_i}$$

We assume that only the owner U_i of the public key P_{k_i} knows how to compute the inverse permutation of f_i efficiently.

Let U_s be the signer who wants to construct the signature on behalf of the group on a message m . The signer has access to a public collision resistant hash function $h()$. During the **RingSign** algorithm the signer uses the extended one-way trap-door permutations (the same way they are described in [21]). The signing procedure will contain the following steps:

1. The signer U_s chooses random $x_i \in \{0, 1\}^*$ such that $i \neq s$ (for all members in the group except himself).
2. The signer computes $g_i(x_i)$ for each chosen x_i , where $g_i(x_i)$ is the extended trap-door one-way permutation for $f_i(x_i)$. He also computes $h(m||\rho)$ where ρ is a random value $\in \{0, 1\}^*$.
3. The signer constructs the polynomial:

$$g_1(x_1) + g_2(x_2) + g_3(x_3) + \dots + g_N(x_N) - h(m||\rho) = 0 \quad (5.1)$$

4. The signer computes x_s by inverting his own one-way permutation $g_s(x_s)$

$$\begin{aligned} g_s(x_s) &= h(m||\rho) - [g_1(x_1) + \dots + g_{s-1}(x_{s-1}) + g_{s+1}(x_{s+1}) + \dots + g_N(x_N)] \\ x_s &= g_s^{-1}(x_s) \end{aligned} \quad (5.2)$$

5. The signer produces the signature σ for message m :

$$\sigma = (P_1, P_2, \dots, P_N, t = 1, x_1, x_2, \dots, x_N) \quad (5.3)$$

The number of signers when $t = 1$ can be excluded from the final construction of the signature. However it will be crucial when $1 < t \leq N$.

The **RingVerify** algorithm verifies the signature σ on a message m by first checking the value of t . If it is not provided the verifier makes the computation for one signer. The verifier checks the equation:

$$z_1 + z_2 + z_3 + \dots + z_N = h(m||\rho) \text{ where } z_i = g_i(x_i), \text{ for } 1 \leq i \leq N \quad (5.4)$$

If the equation is satisfied the verifier accepts the signature and outputs “true”, otherwise rejects the signature and outputs “false”.

We will continue by extending the scheme for $1 < t \leq N$. Let's have two signers U_s and U_r from the group of N members who want to collaborate to sign the message m . The signers choose random x_i 's such that $i \in \{1, \dots, N\} \setminus \{s, r\}$. They construct two polynomials:

$$\begin{aligned} z_1 + z_2 + z_3 + \dots + z_N - h(m||\rho) &= 0 \\ z_1 + 2z_2 + 2^2z_3 + \dots + 2^{N-1}z_N - h(m||\rho) &= 0 \end{aligned} \quad (5.5)$$

where $z_i = g_i(x_i)$ for $1 \leq i \leq N$. Just for simplicity and without loss of generality we will assume that they are the owners of P_1 and P_2 with corresponding g_1 and g_2 one-way functions. The corresponding matrix representation of the system above is:

$$\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} h(m||\rho) - [z_3 + \dots + z_N] \\ h(m||\rho) - [2^2z_3 + \dots + 2^{N-1}z_N] \end{pmatrix} \quad (5.6)$$

The two collaborators have to solve the following system of equations in order to produce the signature on the message m :

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}^{-1} \begin{pmatrix} h(m||\rho) - [z_3 + \dots + z_N] \\ h(m||\rho) - [2^2z_3 + \dots + 2^{N-1}z_N] \end{pmatrix} \quad (5.7)$$

This system can always be solved in terms of z_1 and z_2 because the solutions for these two components depend on the inverse of the Vandermonde matrix which always exists. The owners of P_1 and P_2 provide the solutions for x_1 and x_2 since they are the only two who knows how to invert their one-way functions and the corresponding extended trap-doors. The signature produced for the message m is:

$$\sigma = (P_1, \dots, P_N; t = 2; x_1, \dots, x_N) \quad (5.8)$$

This procedure can be generalized for any number of $t \leq N$ signers. We can assume that these signers are the first t of the N possible signers since that does not have any influence on the result, and it makes the computations a little bit simpler. However, in a real signing procedure, they can be any t users, and the identity of the actual signers should be secret. In this case, we use **TRingSign** algorithm with the following steps:

1. The signers choose $x_i \in_R \{0, 1\}^*$ for $i \in \{1, \dots, N\} \setminus \{1, \dots, t\}$ and $\rho \in \{0, 1\}^*$. They compute $h(m||\rho)$ where m is the message to be signed.
2. They construct the system of t polynomials:

$$\begin{aligned} z_1 + z_2 + z_3 + \dots + z_N &= h(m||\rho) \\ z_1 + 2z_2 + 2^2z_3 + \dots + 2^{N-1}z_N &= h(m||\rho) \\ &\vdots \\ z_1 + tz_2 + t^2z_3 + \dots + t^{N-1}z_N &= h(m||\rho) \end{aligned} \quad (5.9)$$

with a corresponding Vandermonde $t \times t$ system:

$$\begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_t \end{pmatrix} = V_m^{-1} \begin{pmatrix} h(m||\rho) - [z_{t+1} + \dots + z_N] \\ h(m||\rho) - [2^t z_{t+1} + \dots + 2^{N-1} z_N] \\ \vdots \\ h(m||\rho) - [t^t z_{t+1} + \dots + t^{N-1} z_N] \end{pmatrix} \quad (5.10)$$

- Each signer u_i , for $1 \leq i \leq t$, uses the solutions from the previous step z_1, \dots, z_N to calculate his value of x_i , by using his secret key. All calculated x_i 's together with the randomly chosen ones are part of the signature.

The produced signature σ on a message m is:

$$\sigma = (P_{k_1}, P_{k_2}, \dots, P_{k_N}; t; x_1, x_2, \dots, x_N).$$

When the verifier receives the signature σ on a message m , the following steps are performed:

- Check the value of t . If not specified, construct only one equation. Otherwise, construct t -system of equations:

$$\begin{aligned} z_1 + z_2 + z_3 + \dots + z_N &= h(m||\rho) \\ z_1 + 2z_2 + 2^2z_3 + \dots + 2^{N-1}z_N &= h(m||\rho) \\ \vdots & \\ z_1 + tz_2 + t^2z_3 + \dots + t^{N-1}z_N &= h(m||\rho) \end{aligned} \quad (5.11)$$

- Evaluate each equation. If the evaluation process outputs “true” for all of them, the verifier accepts the signature and outputs “true”. Otherwise, the signature is rejected and “false” is being outputted.

5.1 Security of the Vandermonde scheme

A ring signature scheme is said to be secure if it is unforgeable and signer ambiguous. We will show that any efficient adversary \mathcal{A} that can generate with non-negligible probability a valid group signature on a new message m by analyzing $\text{poly}(k)$ group signatures on other chosen messages has to invert t one-way functions f_i on random inputs x_i with non-negligible probability.

Our proof is based on the definition of t -out-of- N unforgeability and anonymity given in Section 2. However, because we are also using hash function as a prevention mechanism

from forgery, we will consider **TRingSign** and **TRingVerify** algorithms that use strong hash function. Additionally, all security arguments that we provide are given in the random oracle model.

Lets recreate the game given in the definition from Section 2. The game described fits the model of adaptive chosen-message attack assuming that whenever \mathcal{A} queries **OSign** it can choose the message adaptively. The adversary \mathcal{A} knows the public keys of all N members of the group. \mathcal{A} also has access to a **CorruptOracle** and has successfully corrupted up to $t - 1$ secret keys S_{i_j} where $i_j \in T$ and $j \leq |T|$. The signing oracle has access to a **HashOracle**. \mathcal{A} can query the signing and the hash oracle $\text{poly}(k)$ times. The adversary \mathcal{A} sends $\text{poly}(k)$ different messages to the signing oracle, for a t -out-of- N -signature. Each message can be adaptively chosen to depend on the response of the previous queries. The adversary will win the game if he can forge a signature on a message that was not queried before. That means he outputs a valid message-signature pair (m, σ) such that **TRingVerify** $(m, \sigma) = "true"$ where m was never before sent as an input to **OSign**.

Theorem 5.1 *If there exists a forger \mathcal{A} which succeeds in breaking the scheme, then one can invert some of the one-way functions f_i .*

Completeness. The signing oracle **OSign** can generate a valid signature given a list of public keys, the threshold t , message m and access to the **HashOracle**.

Soundness. In order to prove the soundness of our scheme, we will introduce two new entities besides the adversary \mathcal{A} : a dealer \mathcal{D} , which will generate an instance of a hard problem to be broken, and a simulator \mathcal{S} which will simulate the signing oracle as well as the hash oracle. The simulated signatures will have the identical distribution as the ones issued by the signing algorithm and the adversary cannot detect any abnormality. If \mathcal{A} wins the following game, then the signature constructed by our scheme can be forged.

The game:

1. \mathcal{D} generates an instance of the hard problem to be broken .
2. \mathcal{S} runs **Gen** (1^k) to generate a list of public keys R . It provides the list to \mathcal{A} altogether with a threshold value t . \mathcal{S} request from \mathcal{A} to deliver (m, σ, ρ) , where m is a message that was not signed before, σ is the corresponding signature and ρ is the random value which is concatenated with the message before the hashing process.

3. \mathcal{A} is allowed to query the signing oracle (simulated by \mathcal{S}) with any message different than the one that is supposed to be delivered to \mathcal{S} . We assume a polynomially bounded adversary which means he can make at most $\text{poly}(k)$ queries to \mathcal{S} .
4. \mathcal{A} delivers the message-signature pair together with ρ .
5. \mathcal{S} uses the response from \mathcal{A} to break the hard problem.

The last step will only be possible if the signature generated by \mathcal{A} is valid and the verification algorithm outputs “true”. That way, \mathcal{S} can extract valid information and break the hard problem. This leads to contradiction which is more visible if we run the routine with a given hard problem. Let say the dealer \mathcal{D} generates an instance: given $(f_1(), h)$ find x_1 such that $f_1(x_1) = h$, assuming that \mathcal{A} did not corrupt S_1 (the adversary has all of the secret keys except for S_1).

\mathcal{S} provides the list of public keys R to \mathcal{A} and the threshold t , and it request σ on a given m and ρ . It also requires that $(f_1(), h)$ be involved in the construction of signature (be part of the threshold). The probability p that \mathcal{A} will successfully deliver requested signature is negligible. From the signature listing the values of x 's \mathcal{S} extracts the value of x_1 and provides to \mathcal{D} . The probability that someone can break the hard problem listed, inverting one-way function, is the same with the probability that \mathcal{A} will successfully forge the signature.

Anonymity Proving the anonymity of this scheme is straightforward. For a given threshold ring signature, each t -set of user could have selected randomly the values x_i for the non-signers, solve the system of equations to find the values z_j for the users in the t -set, and invert those values to find the corresponding values x_j . Hence, any t -set of user could have produced the signature, and the scheme is unconditionally anonymous.

5.2 Extending the Vandermonde scheme to ID-based signatures

Since the introduction of ring signatures, a lot of work has been done to obtain ID based schemes [9, 10, 6, 13, 24]. The scheme that we introduced in the previous section can also be extended for ID based threshold ring signatures. However, this extension was already done by Wei [25]. The construction of the signature provided by Wei is based on Boneh et al's [6] bilinear signature, and it fits the so called spontaneous anonymous groups. These groups

are ad-hoc groups without setup procedure and without group secret. However, there has to exist a binary relation that can be viewed as the group secret of this spontaneous anonymous group.

The scheme consists of two algorithms: **Sign** and **Verify**. Before we explain the process of construction and verification of the signature we must discuss the notion used in the description. Let G_1 and G_2 be two multiplicative cyclic groups of prime order p and each has its own generator g_1 and g_2 respectively, and let ψ be a computational isomorphism from G_2 to G_1 with $\psi(g_2) = g_1$. We assume that the ψ always exists and is efficiently computable. Let e be a computable bilinear map $e: G_1 \times G_2 \rightarrow G_T$ with the following properties:

1. Bilinear: for all $u \in G_1, v \in G_2$ and $a, b \in Z, e(u^a, v^b) = e(u, v)^{ab}$
2. Non-degenerate: $e(g_1, g_2) \neq 1$.

accompanied by two more:

- for any $u_1, u_2 \in G_1, v \in G_2, e(u_1 u_2, v) = e(u_1, v) \cdot e(u_2, v)$, and
- for any $u, v \in G_2, e(\psi(u), v) = e(\psi(v), u)$.

The signing algorithm is based on the existence of a set of possible signers with a set R of public keys $v_i \in R$ such that $v_i = g_2^{x_i}, 1 \leq i \leq N$ and a set of actual signers I such that $I \subset \{1, \dots, N\}$ and $|I| = t$. It is assumed that there are $H_j, 0 \leq j < t$ hashing functions available and each member of I has its own private key x_i . The signature for a given message $m \in \{0, 1\}^*$ is constructed as follows. For each non-member of I , a random value u_i is picked and $\sigma_i = g_1^{u_i}$ is computed. A random tag ρ is picked and a system of t equations is constructed:

$$H_j(m, R, t, \rho) = \left(\prod_{s \in I} \sigma_s^{x_s s^j} \right) \left(\prod_{i \notin I} \psi(v_i)^{u_i i^j} \right) \text{ for } 0 \leq j < t \quad (5.12)$$

This system of equations is solved for each $\sigma_s^{x_s}$ and $s \in I$. The solution is always possible because of the properties of Vandermonde matrices for which there is always an inverse, and there exist two matrices $A^{(I)}$ and $B^{(I)}$ such that:

$$\sigma_s^{x_s} = \left(\prod_{0 \leq j < t} H_j(m, R, t, \rho)^{A_{s,j}^{(I)}} \right) \left(\prod_{i \notin I} \psi(v_i^{u_i})^{B_{s,i}^{(I)}} \right) \quad (5.13)$$

The constructed signature is of the form: $\sigma = (\sigma_1, \dots, \sigma_N, \rho)$. The verifier knows the list of public keys $R = (v_1, \dots, v_N)$, and after receiving the message m with the signature $\sigma \in G_1^N \times R$, he verifies the t equalities as follows:

$$e(H_j(m, R, t, \rho), g_2) = \prod_{i=1}^N e(\sigma_i, v_i^{i_j}) \quad (5.14)$$

As in any other (t, N) scheme, the signature is accepted if all of the equalities are satisfied. Otherwise, the signature is rejected.

CHAPTER 6

Beyond threshold ring signatures: General access structures

In the previous section, we introduced a new scheme for ring signatures based on Vandermonde matrices. We desired that any t out of N participants be able to sign given message m . A more general situation would be if it is specified exactly which subsets of participants can produce a valid signature on a given message, and which subsets cannot. The set of subset that can sign will be specified at the beginning of the signing process. During the verification process the verifier “looks” for a proof that some of the valid subsets signed the message m . To further motivate our effort, we will use our earlier example with the company AB and leaking secrets. Now, we assume that the journalist will not publish the leaked secret unless it is signed by at least one of the following subsets of users: the president of the company; two executive directors; or ten clerks. Our goal is to design a scheme that will convince the journalist that at least one of the specified subsets was involved in the signing. However, the users that leaked the secret want to remain anonymous. Hence, the journalist should not be able to determine which of the specified subsets has generated the signature.

The scheme that we present here is similar to the single user scheme proposed in Section 5. We are going to slightly change the equations that we use to produce the signature. In Section 5, we used equation with N variables. Each variable was calculated by applying one-way trap-door function using the public key of the participants. For the general access structure, each trap-door function will be replaced by a trap-door set-function.

We will call the valid subsets *authorized* and any other subset which cannot produce a signature *forbidden*. Let denote by Γ the set of all authorized subsets and by Δ the set of all forbidden subsets. The tuple (Γ, Δ) is called access structure if $\Gamma \cap \Delta = \emptyset$. If $\Gamma \cup \Delta = 2^N$,

where N is the set of all participants, we say that (Γ, Δ) is complete ¹.

Let $\Gamma = \{U_1, U_2, \dots, U_r\}$, where U_1, U_2, \dots, U_r are authorized subsets. The signature is always constructed from one valid subset U_s . The members of that valid subset have to cooperate in order to produce valid signature. The steps will be as follows:

1. The signing subset U_s chooses a random values $x_i \in \{0, 1\}^*$ such that $i \neq s$ (for every other valid subset except for its own).
2. For each x_i , the signing subset computes $g_{i_1}(g_{i_2} \dots (g_{i_q}(x_i)) \dots) = G(x_i)$, where $q = |U_i|$ and g_{i_j} is the extended trap-door one-way permutation of the j -th member of U_i .
3. The signing subset chooses random tag $\rho \in \{0, 1\}^*$ and computes $h(m||\rho)$ for the message m that is signed.
4. The signing subset constructs the polynomial:

$$G_1(x_1) + G_2(x_2) + \dots + G_s(x_s) + \dots + G_r(x_r) - h(m||\rho) = 0 \quad (6.1)$$

5. The signing subset calculates $G_s(x_s)$ from the polynomial and applies the inversion process of G_s in order to obtain x_s .

$$x_s = g_{i_1}^{-1}(g_{i_2}^{-1} \dots (g_{i_q}^{-1}(G_s(x_s))) \dots) \quad (6.2)$$

This can be done only if all the members from the signing subset collaborate and calculate one step of the inversion process using their own secret key.

6. The valid subset produces the signature σ on a message m the same way that we already described in the Section 5. The only difference is that instead of public keys the valid subsets should be listed as part of the signature.

$$\sigma = (U_1, U_2, \dots, U_r; x_1, \dots x_r) \quad (6.3)$$

When the verifier verifies the signature, he computes each $G_s(x_s)$ and evaluates the polynomial in Equation 6.1. If the resulting value is 0, the message is accepted as valid. Otherwise, the message is rejected. Since any of the subsets could have generated the signature, the verifier cannot do better than to randomly guess the subset of actual signers. Hence, the scheme provides unconditional anonymity.

¹For our purposes we assume that Γ is monotonic. That is if $x \in \Gamma$ and $x' \subset x$ than $x' \in \Gamma$

CHAPTER 7

Efficiency

In this section, we provide efficiency analysis of our schemes and compare them to some of the existing proven secure t -out-of- N schemes.

The basic ring signature scheme [21], when based on Rabin or RSA signatures with public exponent 3, requires one modular exponentiation and a linear number of modular multiplications for each user in the ring during the signing process. It requires at most two modular multiplications for each ring member during the verification process. This means that generating and verifying a ring signature will have approximately the same cost as generating or verifying a regular signature plus an extra multiplications (or two) for each non-signer. However, since the groups are not predefined, the signature itself should contain a list of all ring members. Thus, the size of the signature grows linearly with the size of the ring. This is one of the main disadvantages of ring signatures compared to group signatures.

The BSS threshold t -out-of- N scheme [7] using RSA with exponent 3 requires $\mathcal{O}(N - t)$ modular multiplications, $\mathcal{O}(t)$ modular exponentiations and N polynomial evaluations. To verify such signature $2N$ modular multiplications, and N polynomial evaluations are required. In this case, the size of the signature grows with both, the number of users N and the number of signers t . The size of one such signature is: $\mathcal{O}(l2^t N \log N)$, where t is the number of the signers that participate in generation of the signature, N is the size of the whole set from which the participants are chosen, and l is a security parameter (the size of the input to each trap-door permutation). The scheme is more efficient compared to the trivial solution where all subgroups with cardinality t are listed leading to $\mathcal{O}(N^{t-1})$ time complexity.

Although the BSS scheme is efficient when the number t of signers is small (i.e., $t = \mathcal{O}(\log N)$), it becomes very inefficient when the number of actual signers is large (i.e.,

$t = \omega(\log N)$). An improvement for the case of large sets of signers was proposed by Isshiki and Tanaka. The signing algorithm of their $(N-t)$ -out-of- N threshold ring signature scheme requires $Q \times 2^t \log N$ time for trap-door one-way function inversion, where Q is the size of the partitions, and $\mathcal{O}(tQ2^t N \log N)$ for one-way function evaluation in the easy direction. Hence, the scheme is more efficient when the number of actual signers is close to the number of ring members.

Different ID based threshold ring signatures were proposed in the past 10 years. However, we will not overview their time and size complexity due to the pairing operation which makes these schemes fairly expensive.

The first of our proposed schemes was based on covering designs. For the ring set system that we propose in Section 4, the number of t -sets of users that can generate signature is $(t!)^{d-1}$, and it grows very fast both with t and d . As we already elaborated, each user can be an actual signer, and for each user there is a t -set of users that includes that particular signer and can produce a threshold ring signature. The complexity of the scheme for the given construction set is $\mathcal{O}(d \cdot N) = \mathcal{O}(N \log N)$. If we compare the time complexity of the scheme to the complexity $\mathcal{O}(l2^t N \log N)$ of the Bresson-Stern-Szydlo scheme or the complexity $\mathcal{O}(lt2^t N^2 \log N)$ of the Isshiki-Tanaka scheme, it is not difficult to see that our scheme is significantly more efficient. Namely, the complexity does not increase exponentially with the number of signers t , and it does not increase quadratically with the whole set of possible signers N .

The second scheme that we proposed is based on Vandermonde matrices. In this case, the signing algorithm requires $(N-t)$ evaluations of a trap-door one-way function (the easy direction), and t inversions of a trap-door one-way function (the hard direction). Also, there are $t(N-1)$ modular additions, which can become crucial if the number of signers is large, and t polynomial evaluations. The verifying algorithm requires N evaluations of a trap-door one-way function as well as t polynomial evaluations and $t(N-1)$ modular additions. That means that the complexity of the generated signature is approximately $\mathcal{O}(t \cdot N)$. The size of the signature in this case is $\mathcal{O}(l \cdot t \cdot N)$, and it will grow with the size of the set used for generating signatures. The second scheme is not as efficient as the one based on covering designs when the number of signers is large. However, it achieves perfect anonymity, and is efficient when the number of actual signers is small.

CHAPTER 8

Conclusions

We have proposed two anonymous and unforgeable t -out-of- N signature schemes. The first scheme is based on covering designs, and is unforgeable and unconditionally anonymous. Although the anonymity provided by the scheme is not perfect, this scheme is far more efficient than the existing schemes when the number of actual signers is large. The second scheme, which is based on Vandermonde matrices, is not as efficient as the first one when the number of actual signers is large. However, it provides perfect anonymity. We also present a scheme where the sets of possible signers are not all t -subsets of the group, but defined by a general access structure.

REFERENCES

- [1] N.Alon, R.Yuster, U.Zwick, “Color-coding”. Electronic Colloquium on Computational Complexity(ECCC) 1,009(1994). Full paper appears in J.ACM 42:4, July 1995, pp.844-856. [3.2.1](#), [3.2.1](#)
- [2] N.Asokan, P.Ginzboorg, “Key agreement in Ad-hoc Networks”. In Nordsec '99 Workshop, Krista, Sweden, November 1999. [1](#)
- [3] M.Bellare, D.Micciancio, B.Warinschi, ”Foundations of Group Signatures: Formal Definitions, Simplified Requirements and Construction Based on General Assumption”. In Advances in Cryptology- EUROCRYPT 2003, Warsaw Poland,May 4-8 2003, vol.2656 of Lecture Notes in Computer Science, pp. 614-629, Springer-Verlag. [1](#)
- [4] M.Bellare, H.Shi, C.Zhang, “Foundations of Group Signatures: The Case of Dynamic Groups”. In Topics in Cryptology CT-RSA 2005, San Francisco USA, February 14-18 2005, vol.3376 of Lecture Notes in Computer Science, pp. 136-153, Springer-Verlag. [1](#)
- [5] A.Bender, J.Katz, R.Morselli, “Ring Signatures: Stronger Definitions and Constructions without Random Oracles”. In 3rd Theory of Cryptography Conference, New York USA, March 4-7 2006. [1](#), [2](#)
- [6] D.Boneh, C.Gentry, B.Lynn, H.Shacham, “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps”. In Advances in Cryptology- EUROCRYPT 2003, vol.2656 of Lecture notes in Computer Science, pp. 416-432, Springer-Verlag. [1](#), [5.2](#)
- [7] E.Bresson, J. Stern, M.Szydlo, “Threshold Ring Signatures and Applications to ad-hoc groups”. In Advances in Cryptology-CRYPTO 2002, Santa Barbara, USA, 18-22 August 2002, M. Yung, Ed. vol.2442 of Lecture Notes in Computer Science, pp.465-480, Springer-Verlag. [1](#), [3.2.1](#), [7](#)
- [8] D.Chaum, E.Van Heyst, “Group signatures”. In Advances of Cryptology-EUROCRYPT '91, Brighton, UK, April 1991, D.Davies, Ed. vol.547 of Lecture Notes in Computer Science, pp. 257-265, Springer-Verlag. [1](#)
- [9] S.M.Chow, S.M.Yiu, L.Hui, “Efficient Identity Based Ring Signatures”. In Applied Cryptography and Network Security: Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005, Proceedings,vol.3531 of of Lecture Notes in Computer Science, Springer-Verlag. [5.2](#)

- [10] S.Chow, L.Hui, S.M.Yiu, "Identity Based Threshold Ring Signature". In Information Security and Cryptology - ICISC 2004, 7th International Conference Seoul, Korea, December 2-3, 2004, vol.3506 of Lecture Notes in Computer Science, pp.218-232, Springer-Verlag. [5.2](#)
- [11] Y.Desmedt, K.Kurosawa, "How to break a Practical MIX and Design a new one". In Advances in Cryptology-EUROCRYPT 2000, vol.1807 of Lecture Notes in Computer Science, pp.557-572, Springer-Verlag. [4](#)
- [12] Y.Dodis, A.Kiayias, A.Nicolosi, V.Shoup, "Anonymous Identification in Ad Hoc Groups". In International Conference on the Theory and Applications of Cryptographic Technique - EUROCRYPT 2004, Interlaken, Switzerland, May 2-6 2004, vol.3027 of Lecture Notes in Computer Science, p.p 609-626, Springer-Verlag. [1](#)
- [13] J.Herranz, G.Saez, "New Identity-Based Ring Signature Schemes". In Information and Communication Security, 6-th International Conference, ICICS 2004, Malaga, Spain, October 2004, Proceedings, vol.3269 of Lecture Notes in Computer Science, pp.27-39, Springer-Verlag. [5.2](#)
- [14] T.Isshiki, K.Tanaka, "An (n-t)-out-of-n Threshold Ring Signature Scheme". In Information Security and Privacy, 10th Australasian Conference, ACISP 2005, Brisbane, Australia, July 4-6, 2005, Proceedings, vol.3574 of Lecture Notes in Computer Science, pp.406-416, Springer-Verlag. [1](#), [3.2.2](#)
- [15] M.Ito, A.Saito, T.Nishizeki, "Secret Sharing Scheme Realizing General Access Structure". In Proceedings of IEEE Globecom'87, pp.99-102,1987.
- [16] K. Maneva-Jakimoska, G. Jakimoski and M. Burmester, "Threshold Ring Signatures Efficient for Large Sets of Signers," Cryptology ePrint Archive, 2005/227. [4](#)
- [17] W.Mao, "Modern Cryptography Theory and Practice," Prentice-Hall, 2004.
- [18] W.H.Mills, "Covering design I: coverings by a small number of subsets". Ars Combin.8, (1979), pp.199-315. [4](#)
- [19] C.Perkins, "Ad-hoc Networking," Addison Wesley, 2001 [1](#)
- [20] R.Rees, D.R.Stinson, R.Wei, G.H.J. van Rees, "An application of covering designs: Determining the maximum consistent set of shares in a threshold scheme," Ars Combin.531(1999), pp.225-237. [4](#)
- [21] R.L.Rivest, A.Shamir, Y.Tauman, "How to leak a secret". In Advances in Cryptology-ASIACRYPT 2001, Gold Coast, Australia, December 2001, C.Boyd, Ed., vol.2248 of Lecture Notes in Computer Science, pp.552-565, Springer-Verlag. [1](#), [3.1](#), [3.1](#), [3.1](#), [5](#), [7](#)
- [22] D.R.Stinson, "Cryptography: Theory and Practice," CRC Press,1995.
- [23] K.Tochikubo, T.Uyematsu, R.Matsumoto, "Efficient Secret Sharing Schemes Based on Authorized Subsets". In IEICE Transactions, January 2005,vol.E88-A, pp.322-328.

- [24] B.Waters, “Efficient Identity-Based Encryption Without Random Oracles”. In Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings, vol.3494 of Lecture Notes in Computer Science, pp.114-127, Springer 2005. [5.2](#)

- [25] V.Wei, “A Bilinear Spontaneous Anonymous Threshold Signature for Ad Hoc Groups,” ePrint Cryptology archive, 2004. [5.2](#)

BIOGRAPHICAL SKETCH

Karolina Maneva-Jakimoska

Karolina Maneva-Jakimoska was born and raised in Gevgelija, Macedonia. She has received her Bachelors degree in Electrical Engineering, Electronics and Telecommunications, at Ss. Cyril and Methodius University, Skopje, Macedonia, in April 1997. During the Kosovo crisis she worked for the International Organization for Migration as administrator and database manager for the medical evacuation team. From 1999 to summer 2001, she worked as an instructor in a Technical School in Skopje, Macedonia, teaching a variety of courses in the area of Electronics and Telecommunications. In fall 2001, she moved to Tallahassee Florida, USA. She returned to school in Fall 2003 to pursue a Master of Science Degree on Computer Science Department at Florida State University. During her studies she achieved a Certificate for Information Systems Security Professionals. She is a member of Upsilon Phi Epsilon Computer Science Honor Society, Phi Kappa Phi Honor Society and the Association for Computing Machinery.