MASTER'S THESIS

Master of Science in Advanced Mathematics and Mathematical Engineering

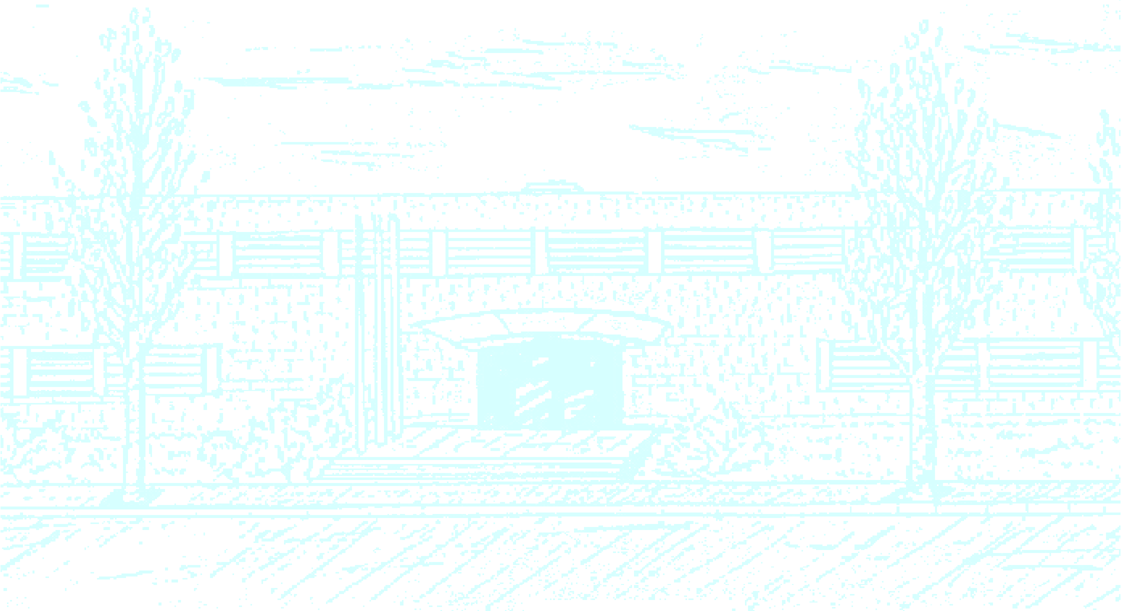
Title: Anonymous threshold signatures

Author: Petar Hlad Colic

Advisor: Javier Herranz Sotoca

Department: Department of Mathematics

Academic year: 2017-2018



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

ANONYMOUS THRESHOLD SIGNATURES

A master's thesis
Submitted to the

Facultat de Matemàtiques i Estadística

Universitat Politècnica de Catalunya

By

Petar Hlad Colic

In partial fulfillment
of the requirements for the master's degree in

Advanced Mathematics and Mathematical Engineering

Advisor: Javier Herranz Sotoca

Barcelona, June 2018

Abstract

Resum

Resum

Resum

Resumen

Resum

ACKNOWLEDGEMENTS

Acknowledgements

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

[PP92] [BS97]

An access structure Γ is the set of all subsets of $\mathcal{P}$ that can recover the secret.

Definition 1.0.1. Let $\mathcal{P} := \{P_1, \ldots, P_n\}$ be a set of participants. A *monotone access structure* Γ on $\mathcal{P}$ is a subset $\Gamma \subseteq 2^{\mathcal{P}}$, which is monotone increasing

$$A \in \Gamma, \quad A \subseteq A' \subseteq \mathcal{P} \Rightarrow A' \in \Gamma$$

Definition 1.0.2. Let $\mathcal{P} := \{P_1, \ldots, P_n\}$ be a set of participants and let $A \subseteq 2^{\mathcal{P}}$. The *closure* of A, denoted $\mathrm{cl}(A)$, is the set

$$\mathrm{cl}(A) = \{C : \exists B \in A \text{ s.t. } B \subseteq C \subseteq \mathcal{P}\}$$

For a monotone access structure Γ we have $\Gamma = \mathrm{cl}(\Gamma)$.

Definition 1.0.3. Let Γ be an access structure on a set of participants $\mathcal{P}$. $B \in \Gamma$ is a *minimal* qualified set if $A \notin \Gamma$ whenever $A \subsetneq B$.

Definition 1.0.4. Let Γ be an access structure on a set of participants $\mathcal{P}$. The family of minimal qualified sets Γ_0 of Γ is called the *basis* of Γ.

For a basis Γ_0 of an access structure Γ we have $\Gamma = \mathrm{cl}(\Gamma_0)$

Definition 1.0.5. An access structure Γ is *trivial* if either $\Gamma = 2^{\mathcal{P}}$ or $\Gamma = \{\mathcal{P}\}$.

Let $\mathcal{K}$ be a set of q elements called *secret keys*, and let $\mathcal{S}$ be a finite set whose elements are called *shares*. Let D be a *dealer* who wants to share a secret key $\mathbf{k} \in \mathcal{K}$ among the participants in $\mathcal{P}$.

Definition 1.0.6. A *distribution rule* is a function $f : \mathcal{P} \cup \{D\} \to \mathcal{K} \cup \mathcal{S}$ which satisfies the conditions $f(D) \in \mathcal{K}$ and $f(P_i) \in \mathcal{S}$ for $i = 1, 2, \ldots, n$.

Secret sharing schemes will be represented by a collection of distribution rules, which represent a possible distribution of shares to the participants where $f(D)$ is the secret key being shared and $f(P_i)$ is the share given to P_i.

Definition 1.0.7. Let $\mathscr{F}$ be a family of distribution rules, and let $\mathbf{k} \in \mathcal{K}$. Then $\mathscr{F}_{\mathbf{k}} := \{f \in \mathscr{F} : f(F) = \mathbf{k}\}$ is the family of all distribution rules having $\mathbf{k}$ as secret.

If $\mathbf{k} \in \mathcal{K}$ is the secret that D wants to share, then D will chose a distribution rule $f \in \mathscr{F}_{\mathcal{K}}$ uniformly at random.

Let $\{p_{\mathcal{K}}(\mathbf{k})\}_{\mathbf{k}\in\mathcal{K}}$ be a probability distribution on $\mathcal{K}$, and let a collection of distribution rules for secrets in $\mathcal{K}$ be fixed.

Definition 1.0.8. A *perfect secret sharing scheme*, with respect to a monotone access structure $\Gamma \subseteq 2^{\mathcal{P}}$, is a collection of distribution rules that satisfy the following two properties:

1. If a subset $A \in \Gamma$ of participants pool their shares, then they can determine the value of the secret $\mathbf{k}$.

2. If a subset $A \notin \Gamma$ of participants pool their shares, then they can determine nothing about the value of the secret $\mathbf{k}$. Formally, if $A \notin \Gamma$ then for all $a = \{(P_i, s_i) : P_i \in A \text{ and } s_i \in \mathcal{S}\}$ with $p(a) > 0$, and for all $\mathbf{k} \in \mathcal{K}$, it holds $p(\mathbf{k}|a) = p_{\mathcal{K}}(\mathbf{k})$. In other words, the *a priori* probability of the value of $\mathbf{k}$ does not change after knowing the shares held by A.

Definition 1.0.9. An *ideal secret sharing scheme* is a secret sharing scheme for which $|\mathcal{K}| = |\mathcal{S}|$. An access structure admitting an ideal secret sharing scheme will be referred as *ideal access structure.*

Theorem 1.0.10. ~~*Let Γ be an access structure on a set of participants $\mathcal{P}$. An ideal anonymous secret sharing scheme for Γ exists if and only if either Γ is a $(1, |\mathcal{P}|)$ threshold structure, a $(|\mathcal{P}|, |\mathcal{P}|)$ threshold structure, or the closure of a complete bipartite graph.*~~

CHAPTER 2

PRELIMINARIES

Some cryptographic preliminaries and other definitions.

2.1 Bilinear pairings

[DH76]

Let G_1 and G_2 be two (multiplicative) cyclic groups of prime order q. Let g_1 be a fixed generator of G_1 and g_2 be a fixed generator of G_2.

Definition 2.1.1. Computation Diffie-Hellman (CDH) Problem: Given a randomly chosen $g \in G_1$, g^a, and g^b (for unknown randomly chosen $a, b \in \mathbb{Z}_q$), compute g^{ab}.

Definition 2.1.2. Decision Diffie-Hellman (DDH) Problem: Given randomly chosen $g \in G_1$, g^a, g^b, and g^c (for unknown randomly chosen $a, b, c \in \mathbb{Z}_q$), decide whether $c = ab$. (If so, (g, g^a, g^b, g^c) is called a valid Diffie-Hellman tuple.)

Definition 2.1.3. Computational co-Diffie-Hellman (co-CDH) Problem on (G_1, G_2): Given $g_2, g_2^a \in G_2$ and $h \in G_1$ as input, compute $h^a \in G_1$.

Definition 2.1.4. Decision co-Diffie-Hellman (co-DDH) on (G_1, G_2): Given $g_2, g_2^a \in G_2$ and $h, h^b \in G_1$ as input, decide whether $a = b$. If so, we say that (g_2, g_2^a, h, h^a) is a co-Diffie-Hellman tuple.

Definition 2.1.5. Bilinear map: Let G_T be an additional group such that $|G_1| = |G_2| = |G_T|$. A bilinear map is a map $e : G_1 \times G_2 \to G_T$ with the following properties:

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

Definition 2.1.6. A Gap co-Diffie-Hellman (co-GDH) group pair is a pair of groups (G_1, G_2) on which co-DDH is easy but co-CDH is hard. When (G_1, G_1) is a co-GDH group pair, we say G_1 is a Gap group (GDH).

Remark 2.1.7. If there s a bilinear map on G_1, G_2, then they are a co-GDH group pair. If there is a bilinear map over $G_1 \times G_1$, then G_1 is a gap group, since one can use the bilinear map to solve the DDH problem.

2.2 Oblivious Polynomial Evaluation

Oblivious polynomial evaluation is a protocol involving two parties, a sender whose input is a polynomial P, and a receiver whose input is a value α. At the end of the protocol, the receiver learns $P(\alpha)$ and the sender learns nothing.

Generic Protocol:

- Input:
  - Sender: a polynomial $P(y) = \sum_{i=0}^{d_P} b_i y^i$ of degree d_P in the field $\mathbb{F}$.
  - Receiver: a value $\alpha \in \mathbb{F}$

- Output:
  - Sender: nothing.
  - Receiver: $P(\alpha)$
- Protocol security parameters: m, k.

Protocol:

1. The sender hides P in a bivariate polynomial:

The sender generates a random masking polynomial $P_x(x)$ of degree d s.t. $P_x(0) = 0$.

$$P_x(x) = \sum_{i=1}^{d} a_i x^i$$

where $d = k \cdot d_P$.

The sender defines a bivariate polynomial

$$Q(x, y) = P_x(x) + P(y) = \sum_{i=1}^{d} a_i x^i + \sum_{i=0}^{d_P} b_i y^i$$

2. The receiver hides α in a univariate polynomial:

The receiver chooses a random polynomial S of degree k, such that $S(0) = \alpha$.

Define $R(x) = Q(x, S(x))$.

[NP99]

2.3 Homomorphic PKE

2.4 Digital Signatures

To ensure integrity of data in communications and authentication, the concept of digital signatures was developed.

A digital signature scheme consists of 3 algorithms:

- **Key generation**: on input of a security parameter k (usually the desired length for the keys), outputs a pair (sk, pk) of secret and public keys.
- **Signature**: given an input message m and the secret key sk, outputs a signature σ.
- **Verification**: given an input message m, a signature σ on the message and a public key pk, outputs whether the signature is valid or not.

A signature scheme must satisfy the following properties:

- **Correctness**: A signature generated with the signing algorithm must always be accepted by the verifier.
- **Unforgeability**: Only a user can sign messages on behalf of himself.
- **Non-repudiation**:

2.4.1 Examples

ElGamal

[ElG85] Let H be a collision-resistant hash function. Let p be a large prime such that the *discrete logarithm problem* is difficult over $\mathbb{Z}_p$. Let g be a randomly chosen generator of $\mathbb{Z}_p^*$

Key generation. Randomly choose a secret key $x \in \mathbb{Z}_p^*$, and compute the public key $y = g^x$.

Signature. To sign a message m, the signer chooses a random $k \in \mathbb{Z}_p^*$. Compute $r = g^k$. To compute s, the following equation must be satisfied: $g^{H(m)} = g^{xr} g^{ks}$. So $s = (H(m) - xr)\, k^{-1} \pmod{p-1}$

If $s = 0$, it starts over again with a different k.

The pair (r, s) is the digital signature for m.

Verification. Check $g^{H(m)} = y^r r^s$

The use of $H(\cdot)$ prevents an existential forgery attack.

Boneh-Lynn-Shacham (BLS)

[BLS01] Let G_1, G_2 be a bilinear group pair of prime order p. Let g be a generator of G_1. Let $e : G_1 \times G_2 \rightarrow G_T$ be a non-degenerate bilinear pairing. Let $H : \{0,1\}^* \rightarrow G_1$ be a full-domain hash function.

Key generation. Randomly choose a secret key $x \in \mathbb{Z}_p$. The public key is $y = g_2^x$.

Signature. Given a private key $x \in \mathbb{Z}_p$, and a message $m \in \{0,1\}^*$, compute $h = H(m) \in G_1$ and $\sigma = h^x$. The signature is $\sigma \in G_1$.

Verification. Given a public key y, a message $m \in \{0,1\}^*$ and a signature $\sigma \in G_1$, compute $h = H(m) \in G_1$ and verify that $e(\sigma, g_2) = e\,(h, y)$.

Schnorr

[Sch90] Let G be a cyclic group of prime order p. Let g be a generator of G. Let $H : \{0,1\}^* \to \mathbb{Z}_p$ be a hash function.

Key generation. Randomly choose a secret key $x \in \mathbb{Z}_p$. The public key will be $y = g^x$.

Signature. Randomly choose $z \in \mathbb{Z}_p$, and compute $L := g^z$.
Compute $c := H(L \parallel m)$.
Compute $s := z + c \cdot x$
The signature on m is $\sigma = (c, s)$.

Verification. Given a signature σ and a public key y, computes $L^\dagger := g^s y^{-c}$ and then check that $c = H(L^\dagger \parallel m)$

2.5 Signature Aggregation

Explain how different signature schemes allow aggregation of n signatures on n messages from n signers.

2.6 Group Signatures

Use of aggregation: group signatures. They are used to sign on behalf of the group, prove group membership.

The easiest way is to give everyone the secret key, so they can sign. But this would let any colluded user to share the secret key to other parties, which is not admissible.

Some group signatures need what is called a Dealer, which will deal with the keys.

Examples of Group signatures:

2.7 Shamir Secret Sharing

Shamir secret sharing described in [Sha79].

The scheme is based on polynomial interpolation.

Let p be a large prime number. All operations are done in $\mathbb{Z}_p$

We want to share a secret s into n shares so that the secret can be recovered with any k distinct shares.

Let $q(x) = a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}$ be a random polynomial of degree $k-1$ in which $a_0 = s$.

Each participant i in $\mathcal{P} = \{1, ..., n\}$ is given a different random number $x_i \in \mathbb{Z}_p$ which identifies the participant. Then, each participant i is given the share $y_i = q(x_i)$.

To recover the secret, we only need k different shares. Let $P \subset \mathcal{P}$ be any subset of k participants. Then

Let $\lambda_i^P = \prod_{P_j \in (P \setminus P_i)} \frac{-x_j}{x_i - x_j}$

$$q(x) = \sum_{i \in P} y_i \prod_{\substack{j \in P \\ i \neq j}} \frac{x - x_j}{x_i - x_j}$$

Then, $s = q(0)$.

2.8 Threshold signature scheme using Shamir

This signature scheme will be based on the BLS scheme.

Let G be a gap group of some prime order p. Let $g \in G$ be a generator of the group.

Let $\mathcal{P} = \{1, \ldots, n\}$ be the set of participants in this scheme. Suppose every participant $P_i \in \mathcal{P}$ is given a distinct random number $x_i \in \mathbb{Z}_p$ as in the Shamir Secret Sharing Scheme.

Let $SK \in \mathbb{Z}_p$ be the secret key of the scheme. Let $q(x) = \sum_{i=0}^{t-1} a_i x^i$ be a random polynomial of degree $t - 1$ but fixing $a_0 = SK$.

Each participant P_i is given the share $s_i = q(x_i)$.

Then, for any set $P \subseteq \mathcal{P}$ of t participants

$$SK = \sum_{i \in P} s_i \lambda_i^P$$

To sign a message m, each participant P_i computes his partial signature $\sigma_i(m) = H(m)^{s_i}$ and broadcasts the pair $(x_i,\ \sigma_i(m))$.

Then, after a set P of at least t participants has broadcast their partial signatures for the message, a standard signature σ can be computed:

$$\sigma(m) = \prod_{P_i \in P} \sigma_i(m)^{\lambda_i^P} = H(m)^{\sum_{P_i \in P} \lambda_i^P s_i} = H(m)^{SK}$$

The signature is valid if $e(\sigma, g) = e(H(m), PK)$

2.9 Anonymity

In order to compare different schemes we need to clear up the definition of anonymity.

The word anonymity is derived from the Greek word *anonymia*, meaning "without a name". In technical terms, the "name" of a participant would be something that uniquely identifies him, e.g. his public key. So, a scheme would be anonymous if the public key of the participant is not disclosed or cannot be obtained in any way at any moment.

It is not very intuitive how we can punt the concept of anonymity in a signature scheme. An anonymous signature does not make much sense. There is no use of an information signed by an anonymous person. What is useful is a signature from a known group of people, but cannot determine from which one nor distinguish two signatures of different participants of the same group on the same message.

- Absolute anonymity. No use. Same as no signature - Anonymity inside a group.

You could use ring signatures which proofs the knowledge of a 1-out-of-N secret key. This could be useful for a small amount of signatures. But it is not useful to provide general anonymity.

A signature scheme provides anonymity if:

- Unlinkability: cannot decide whether two different signatures were signed by the same user.
- Untraceability: cannot get the public key of the signer from a valid signature.

[BS97]

An ideal secret sharing scheme is a scheme in which the size of the shares given to each participant is equal to the size of the secret.

In an anonymous secret sharing schemes the secret can be reconstructed without the knowledge of which participants hold which shares.

CHAPTER 3

SINGLE USE: ANONYMITY

To identify someone is to find his public key.

We can easily achieve anonimity using "pseudonyms", but usually there is a dealer which is the trusted party. So, the security relies on a single party.

The idea:

Let there be a votation with l choices $Z = z_1, \ldots, z_l$. The candidate z_i needs at least t votes to be validated.

The system chooses a random polynomial $q_i(x) = a_{0,i} + a_{1,i}x + \cdots + a_{k-1,i}x^{k-1}$ of degree $k-1$ for each candidate, where $a_{0,i} = SK_i$ is the secret that validates each candidate.

CHAPTER 4

MULTIPLE USE: ANONYMITY AND NON-TRACEABILITY

To achieve full anonymity we need some Multiple use

4.1 Linear size anonymous threshold signature

[DDSV09]

Based on Shamir's secret sharing scheme, this will set a (t, n)-threshold secret sharing scheme.

Setup Algorithm

Let $\mathcal{P} = \{P_1, \dots, P_n\}$ be the set of participants.

The set of participants is divided into r groups $\mathcal{P}_1, \dots, \mathcal{P}_r$, with each group consisting of n/r participants ($\mathcal{P}_i = \{P_1^i, \dots, P_{n/r}^i\}$). For simplicity, we assume that $r|n$.

Let $q > n$ be a sufficiently large prime. Let $a_0, \dots, a_{t-1} \in_R \mathbb{Z}_q$, where $s = a_0 \in \mathbb{Z}_q$ is the secret key.

We set up a (t, r)-threshold signature scheme.

When t participants try to sign a message, there is a probability $p_{\text{fail}} = 1 - p_{\text{success}}$ that at least two participants belong to the same group. The probability that all t participants belong to different groups is given by $p_{\text{success}} = \frac{\binom{r}{t}\left(\frac{n}{r}\right)^t}{\binom{n}{t}}$.

Looks like, for large values, $p_{\text{fail}} \sim \frac{t^2}{2r}$

In [DDSV09] they extend this scheme in the following way: There are set d different (t, r)-threshold signature scheme and every participant is given d shares, one for each secret. The probability of not being able to sign a message is $(p_{\text{fail}})^d$.

To be anonymous, we need $n \geq 2r$.

These schemes would be suitable for small values of t.

Some values:

$\frac{t^2}{2r}$	t	$\frac{n}{r}$	n	r	p_{fail}
10^{-3}	5	10^3	$12.5 \cdot 10^6$	$12.5 \cdot 10^3$	$0.80 \cdot 10^{-3}$
	10		$50 \cdot 10^6$	$50 \cdot 10^3$	$0.90 \cdot 10^{-3}$
	50		$1.25 \cdot 10^9$	$1.25 \cdot 10^6$	$0.98 \cdot 10^{-3}$
	100		$5 \cdot 10^9$	$5 \cdot 10^6$	$0.99 \cdot 10^{-3}$

[DSW04]

4.2 Non-linear size anonymous threshold signature

[CNW11]

Setup Algorithm

Let G_1, G_2, G_T be cyclic groups of sufficiently large prime order q. Two random generators $g_1 \in G_1$, $g_2 \in G_2$, and a bilinear pairing $\hat{t} : G_1 \times G_2 \to G_T$.

DDH problem in G_1, Gap-DL problem in G_1 and G_2 and the blind bilinear LRSW problem are hard.

Let $H_0 : \{0,1\}^* \to \mathbb{Z}_q$ and $H_1 : \{0,1\}^* \to G_1$ be two hash functions.

For each issuer $i \in \mathcal{I}$ the following is performed.

Two integers are selected $x, y \in_R \mathbb{Z}_q$ and the issuer secret key **isk** is assigned to be (x, y). Then the values $X = g_2^x \in G_2$ and $Y = g_2^y \in G_2$ are computed. The issuer public key **ipk** is assigned to be (X, Y).

Finally the system public parameters par are set to be $par = (G_1, G_2, G_T, \hat{t}, g_1, g_2, H_0, H_1, \mathrm{ipk}_k)$ and are published.

Join protocol

This is a protocol between a given signer $s \in S$ and an issuer $i \in \mathcal{I}$.

(Maybe could be a random $f \in G_1$) The signer generates a secret value f using its internal seed **TAAseed**, along with the value $\mathbf{K}_\mathrm{I}$ provided by i and a count number **cnt**.

Signer($\mathfrak{s}$)		Issuer($\mathfrak{i}$)
$f \in_R \mathbb{Z}_q,\ F = g_1^f$		
str $\leftarrow X \parallel Y \parallel n_I$	$\xleftarrow{\text{comm}_{\text{req}}}$	$\text{comm}_{\text{req}} \leftarrow n_I$
	$\longrightarrow$	If $F = g_1^{f_i}$ for any f_i on the roge list then **abort**
		$r \in_R \mathbb{Z}_q;\ A = g_1^r;\ B = A^y$
If $\hat{t}(A, Y) \neq \hat{t}(B, g_2)$	$\xleftarrow{\text{cre}}$	$C = (A^x \cdot F^{rxy});\ cre \leftarrow (A, B, C)$
or $\hat{t}(A \cdot B^f, X) \neq \hat{t}(C, g_2)$		
then **abort**		

Figure 4.1: The Join Protocol

```
Signer($\mathfrak{s}$)
Input: $f \in \mathbb{Z}_q;\ n_T \leftarrow \{0,1\}^*;\ msg$
$a \in_R \mathbb{Z}_q;\ z \in_R \mathbb{Z}_q$
$J \leftarrow H_1(\text{msgt});\ K = J^f;\ L = J^z$
$R = A^a;\ S = B^a;\ T = C^a;\ \tau = \hat{t}(S, X)^z$
$c \leftarrow H_0(R \parallel S \parallel T \parallel \tau \parallel J \parallel K \parallel L \parallel n_T \parallel \text{msgb})$
$s \leftarrow z + c \cdot f \mod q$
$\sigma \leftarrow (R, S, T, J, K, c, s, n_T)$
Output: $\sigma$
```

Figure 4.2: The Sign Algorithm

```
Verifier($\mathfrak{v}$)
Input: $\mathbf{ipk}_k = (X, Y);\ \text{msg} = (\text{msgt}, \text{msgb})$
$\sigma = (R, S, T, J, K, c, s, n_T)$
If $K = J^{f_i}$, for any $f_i$in the set of rogue secret keys, or
    $\hat{t}(R, Y) \neq \hat{t}(S, g_2)$, or
    $J \neq H_1(\text{msgt})$ return reject
$\rho_a^\dagger = \hat{t}(R, X); \rho_b^\dagger = \hat{t}(S, X); \rho_c^\dagger = \hat{t}(T, g_2)$
$\tau^\dagger = (\rho_b^\dagger)^s \cdot (\rho_c^\dagger / \rho_a^\dagger)^{-c}$
$L^\dagger = J^s \cdot K^{-c}$
If $c \neq H_0(R \| S \| T \| \tau^\dagger \| J \| K \| L^\dagger \| n_T \| \text{msgb})$ return reject
Otherwise return accept
```

Figure 4.3: The Verify Algorithm

In the Sign Algorithm (Fig. 4.2), the computation of L, c and s is for the non-interactive Zero-knowledge proof of knowledge of f, and τ is to proof the knowledge of the issuer's credentials. This is based on the *Schnorr protocol* described in [Sch90]

To see if two signatures with same message title were signed by the same signer, only have to check whether $J_0 = J_1$ and $K_0 = K_1$.

In the case of using this scheme in a polling system, **msgt** could be used to identify the poll and **msgb** for the poll choice.

4.3 Anonymous interactive protocol

We propose an interactive protocol based on the signature scheme described in section 2.8. Recall that this scheme does not have the *unlinkability* property because it uses "pseudonyms" and they are shared to compute the signature. Thus, the idea of this new protocol is to hide these "pseudonyms".

As in the **¿previous?** scheme, a participant P_i from a subset $P \subset \mathcal{P}$ of t participants will compute a partial signature $\sigma_i(m)$ on a message m. The partial signature will be $\sigma_i(m) = H(m)^{s_i \prod_{P_j \in (P \setminus P_i)} \frac{-\alpha_j}{\alpha_i - \alpha_j}}$.

The goal of this modification is to compute $a^{\frac{-\alpha_j}{\alpha_i - \alpha_j}}$ without sharing the values of α_i and α_j, for $1 \neq a \in G$ and a given $P_j \in P$. ~~The protocol needs t^2 interactions as the one described in figure **??** to be able to compute the whole signature.~~

To compute $a^{\frac{-\alpha_j}{\alpha_i - \alpha_j}}$ we need participants P_i, P_j and a third party P_s that could be any other participant or a reliable party (like secure hardware).

4.3.1 Description

Interaction

Let $a^{\frac{-\alpha_j}{\alpha_i - \alpha_j}} \leftarrow \mathcal{B}(a, P_i, P_j)$ the protocol that outputs $a^{\frac{-\alpha_j}{\alpha_i - \alpha_j}}$ given $1 \neq a \in G$, a first participant P_i and a second participant P_j.

This protocol is split in five steps. The description follows:

First step: P_i chooses $x_i, x_j, x_s \in \mathbb{Z}_p^*$ three random values and shares them with P_j and P_s. These will be the new "pseudonyms".

P_i and P_j randomly choose polynomials f_i, g_i, z_i and f_j, g_j, z_j, respectively, where: f_i, f_j and g_i, g_j are linear, $g_i(0) = \alpha_i$ and $g_j(0) = \alpha_j$, and z_i, z_j are quadratic polynomials with $z_i(0) = z_j(0) = 0$.

Second step: Let $h(x) = (g_i(x) + g_j(x))(f_i(x) - f_j(x)) + z_i(x) + z_j(x)$. Note that $h(0) = v \cdot (\alpha_i - \alpha_j)$ for $v := g_i(0) + g_j(0)$.

P_i and P_j share with the rest the evaluations of the random polynomials s.t. P_i, P_j, P_s can compute the evaluations $h(x_i), h(x_j), h(x_s)$ respectively.

Third step: P_i,P_j,P_s compute the evaluation of h and share it with the rest. P_i and P_j interpolate the value $h(0)$.

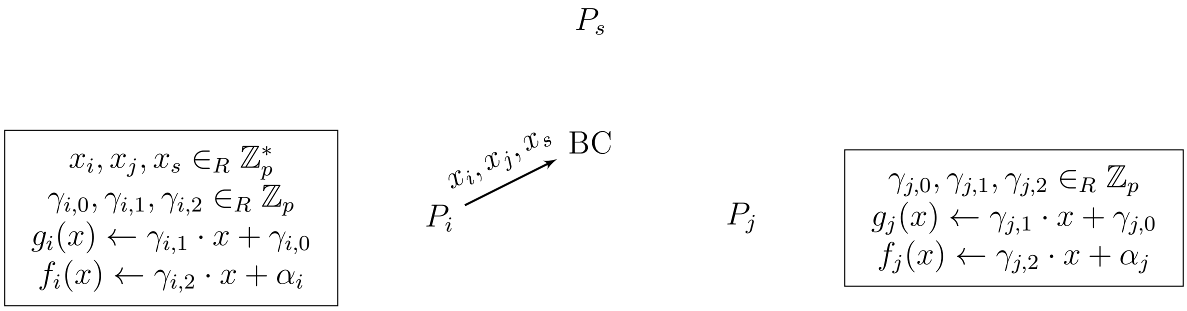

P_s
$x_i, x_j, x_s \in_R \mathbb{Z}_p^*$
$\gamma_{i,0}, \gamma_{i,1}, \gamma_{i,2} \in_R \mathbb{Z}_p$
$g_i(x) \leftarrow \gamma_{i,1} \cdot x + \gamma_{i,0}$
$f_i(x) \leftarrow \gamma_{i,2} \cdot x + \alpha_i$
x_i, x_j, x_s
BC
P_i
P_j
$\gamma_{j,0}, \gamma_{j,1}, \gamma_{j,2} \in_R \mathbb{Z}_p$
$g_j(x) \leftarrow \gamma_{j,1} \cdot x + \gamma_{j,0}$
$f_j(x) \leftarrow \gamma_{j,2} \cdot x + \alpha_j$

Figure 4.4: Step 1

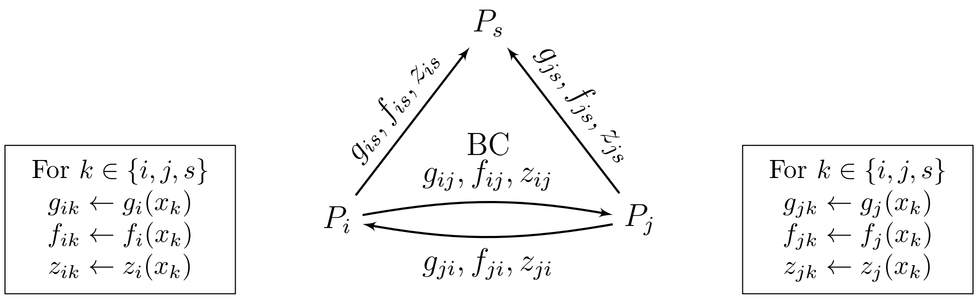

P_s
g_{is}, f_{is}, z_{is}
g_{js}, f_{js}, z_{js}
For $k \in \{i, j, s\}$
$g_{ik} \leftarrow g_i(x_k)$
$f_{ik} \leftarrow f_i(x_k)$
$z_{ik} \leftarrow z_i(x_k)$
BC
g_{ij}, f_{ij}, z_{ij}
P_i
P_j
g_{ji}, f_{ji}, z_{ji}
For $k \in \{i, j, s\}$
$g_{jk} \leftarrow g_j(x_k)$
$f_{jk} \leftarrow f_j(x_k)$
$z_{jk} \leftarrow z_j(x_k)$

Figure 4.5: Step 2

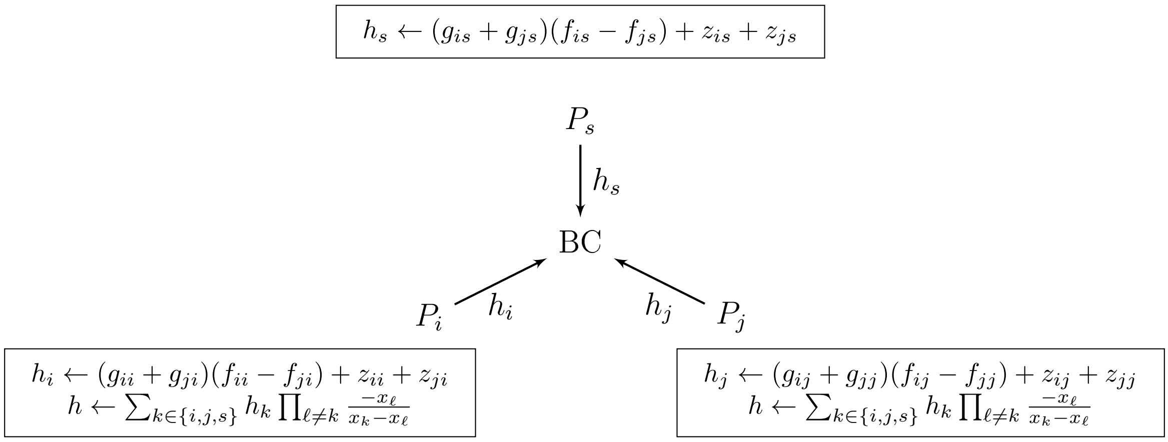

$h_s \leftarrow (g_{is} + g_{js})(f_{is} - f_{js}) + z_{is} + z_{js}$
P_s
h_s
BC
P_i
h_i
h_j
P_j
$h_i \leftarrow (g_{ii} + g_{ji})(f_{ii} - f_{ji}) + z_{ii} + z_{ji}$
$h \leftarrow \sum_{k \in \{i,j,s\}} h_k \prod_{\ell \neq k} \frac{-x_\ell}{x_k - x_\ell}$
$h_j \leftarrow (g_{ij} + g_{jj})(f_{ij} - f_{jj}) + z_{ij} + z_{jj}$
$h \leftarrow \sum_{k \in \{i,j,s\}} h_k \prod_{\ell \neq k} \frac{-x_\ell}{x_k - x_\ell}$

Figure 4.6: Step 3

Fourth step: P_i,P_j compute $A_i = a^{\frac{1}{h(0)}(g_i(x_i)+g_j(x_i))}$, $A_j = a^{\frac{1}{h(0)}(g_i(x_j)+g_j(x_j))}$ respectively. P_j shares A_j with P_i.

P_j can interpolate the exponents of A_i and A_j and compute $a^{\frac{g_i(0)+g_j(0)}{h(0)}} = a^{\frac{v}{v(\alpha_i-\alpha_j)}} = a^{\frac{1}{\alpha_i-\alpha_j}}$.

Fifth step: P_j computes $B^{-\alpha_j} = a^{\frac{-\alpha_j}{\alpha_i-\alpha_j}}$ and shares it with P_i.

Partial signature

Let $P = \{P_i, P_{j_1}, \ldots, P_{j_{t-1}}\}$.

For P_i to compute the partial signature over a message m, computes $\sigma_i(m) = (a_{t-1})^{s_i}$ where $a_k \leftarrow \mathcal{B}(a_{k-1}, P_i, P_{j_k})$ for $k \in \{1, ..., t-1\}$ and $a_0 = H(m)$

Signature

The signature $\sigma(m)$ on a message m from a group of t participants $P = \{P_1, ..., P_t\}$ is

$$\sigma(m) = \prod_{i=1}^{m} \sigma_i(m)$$

.

$$\sigma_i(m) = {a_{t-1}}^{\frac{-\alpha_{j_{t-1}}}{\alpha_i-\alpha_{j_{t-1}}}} = {a_k}^{\frac{-\alpha_{j_k}}{\alpha_i-\alpha_{j_k}} \cdots \frac{-\alpha_{j_{t-1}}}{\alpha_i-\alpha_{j_{t-1}}}} = {a_1}^{\frac{-\alpha_{j_1}}{\alpha_i-\alpha_{j_1}} \cdots \frac{-\alpha_{j_{t-1}}}{\alpha_i-\alpha_{j_{t-1}}}} = H(m)^{s_i \frac{-\alpha_{j_1}}{\alpha_i-\alpha_{j_1}} \cdots \frac{-\alpha_{j_{t-1}}}{\alpha_i-\alpha_{j_{t-1}}}}$$

4.3.2 Analysis

Correctness

Interaction:

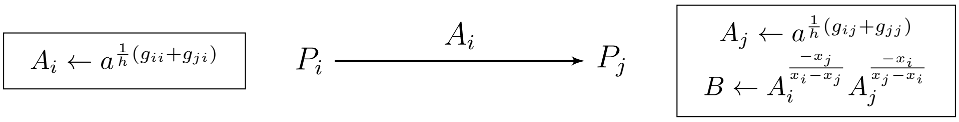

$A_i \leftarrow a^{\frac{1}{h}(g_{ii}+g_{ji})}$
P_i
A_i
P_j
$A_j \leftarrow a^{\frac{1}{h}(g_{ij}+g_{jj})}$
$B \leftarrow A_i^{\frac{-x_j}{x_i-x_j}} A_j^{\frac{-x_i}{x_j-x_i}}$

Figure 4.7: Step 4

$$\begin{tikzcd}
\boxed{\text{Output: } B'} & P_i & P_j \arrow[l, "B'"'] & \boxed{B' \leftarrow B^{-\alpha_j}}
\end{tikzcd}$$

Figure 4.8: Step 5

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

Conclusions and future work

BIBLIOGRAPHY

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. “Short Signatures from the Weil Pairing”. In: *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*. ASIACRYPT ’01. London, UK, UK: Springer-Verlag, 2001, pp. 514–532. ISBN: 3-540-42987-5. URL: http://dl.acm.org/citation.cfm?id=647097.717005.

[BS97] C. Blundo and D.R. Stinson. “Anonymous secret sharing schemes”. In: *Discrete Applied Mathematics* 77.1 (1997), pp. 13 –28. ISSN: 0166-218X. DOI: https://doi.org/10.1016/S0166-218X(97)89208-6. URL: http://www.sciencedirect.com/science/article/pii/S0166218X97892086.

[CNW11] Liqun Chen, Siaw-Lynn Ng, and Guilin Wang. “Threshold Anonymous Announcement in VANETs”. In: *IEEE Journal on Selected Areas in Communications* 29.13 (2011), pp. 605–615. DOI: 10.1109/JSAC.2011.110310.

[DDSV09] Vanesa Daza, Josep Domingo-Ferrer, Francesc Sebé, and Alexandre Viejo. “Trustworthy Privacy-Preserving Car-Generated Announcements in Vehicular Ad Hoc Networks”. In: *IEEE Transactions on Vehicular Technology* 58.4 (2009), pp. 1876–1886. DOI: 10.1109/TVT.2008.2002581.

[DH76] W. Diffie and M. E. Hellman. “New Directions in Cryptography”. In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654.

[DSW04] D. Deng, D. R. Stinson, and R. Wei. “The Lovász Local Lemma and Its Applications to some Combinatorial Arrays”. In: *Designs, Codes and Cryptography* 32.1 (2004), pp. 121–134. ISSN: 1573-7586. DOI: 10.1023/B:DESI.0000029217.97956.26. URL: https://doi.org/10.1023/B:DESI.0000029217.97956.26.

[ElG85] Taher ElGamal. “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”. In: *Advances in Cryptology*. Ed. by George Robert Blakley and David Chaum. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 10–18. ISBN: 978-3-540-39568-3.

[NP99] Moni Naor and Benny Pinkas. “Oblivious Transfer and Polynomial Evaluation”. In: *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*. STOC ’99. Atlanta, Georgia, USA: ACM, 1999, pp. 245–254. ISBN: 1-58113-067-8. DOI: 10.1145/301250.301312. URL: http://doi.acm.org/10.1145/301250.301312.

[PP92] Steven J. Phillips and Nicholas C. Phillips. “Strongly ideal secret sharing schemes”. In: *Journal of Cryptology* 5.3 (1992), pp. 185–191. ISSN: 1432-1378. DOI: 10.1007/BF02451114. URL: https://doi.org/10.1007/BF02451114.

[Sch90] C. P. Schnorr. “Efficient Identification and Signatures for Smart Cards”. In: *Advances in Cryptology — CRYPTO’ 89 Proceedings*. Ed. by Gilles Brassard. New York, NY: Springer New York, 1990, pp. 239–252. ISBN: 978-0-387-34805-6.

[Sha79] A. Shamir. “How to Share a Secret”. In: *Communications of the ACM* 22.11 (1979), pp. 612–613.