

# Anonymous Threshold Signatures

Petar Hlad Colic

Universitat Politècnica de Catalunya  
Facultat de Matemàtiques i Estadística

July 2018

1 Introduction

2 Preliminaries

3 Anonymity

4 Single Use: Anonymity

5 Multiple Use

6 Conclusions

# Goals of the work

- Find efficient anonymous threshold signature scheme with compact signature

# Examples

- Case 1 Toll pricing system that gives discount if there are at least three passengers in vehicle.
- Case 2 E-voting system where each candidate needs certain amount of signatures to get to next round.
- Case 3 Advertising company pays website holders for showing ads, but only pay on the amount of distinct users receiving the ads.

Requirements for these schemes:

- All anonymous
- For case 1: compact signatures, not excessively complex to compute. Could be interactive.
- For case 2: Non interactive.
- For case 3: Compact and non interactive.

We want these schemes to be anonymous.

# PKE scheme

A public key encryption scheme  $PKE = (KG, \mathcal{E}, \mathcal{D})$  consists of three probabilistic and polynomial time algorithms:

- Key generation  $KG$ :
  - Input: Security parameter
  - Output: Pair  $(sk, pk)$  of secret and public keys.
- Encryption  $\mathcal{E}$ :
  - Input: Plaintext  $m$
  - Output: Ciphertext  $c = \mathcal{E}_{pk}(m)$
- Decryption  $\mathcal{D}$ :
  - Input: Ciphertext  $c$
  - Output: Plaintext  $m = \mathcal{D}_{sk}(c)$

For any pair  $(sk, pk)$  and any plaintext  $m$ , it must hold

$$m = \mathcal{D}_{sk}(\mathcal{E}_{pk}(m))$$

# Homomorphic PKE

## Definition 2.1 (Homomorphic PKE).

*Let  $\mathcal{M}$  be the set of plaintexts s.t. it is closed under an operation  $\bullet$ . Let  $\mathcal{C}$  be the set of ciphertexts s.t. it is closed under an operation  $\circ$ . A PKE scheme  $(KG, \mathcal{E}, \mathcal{D})$  has the homomorphic property if*

$$\mathcal{D}_{sk}(\mathcal{E}_{pk}(m_1) \circ \mathcal{E}_{pk}(m_2)) = m_1 \bullet m_2 \quad \forall m_1, m_2 \in \mathcal{M}.$$

## Remark 2.2.

*If we write  $\mathcal{M}$  additively and  $\mathcal{C}$  multiplicatively, for  $a \in \mathbb{Z}^+$  we have:*

$$\mathcal{D}_{sk}(\mathcal{E}_{pk}(m)^a) = a \cdot m$$

# Oblivious Polynomial Evaluation

**Oblivious Polynomial Evaluation** is a protocol involving a sender who knows a polynomial  $P \in \mathbb{F}[x]$  and a receiver who knows a value  $\alpha \in \mathbb{F}$ . At the end of the protocol, the receiver learns  $P(\alpha)$  and the sender learns nothing.

	Sender	Receiver
Input	$P \in \mathbb{F}[x]$	$\alpha \in \mathbb{F}$
Output	-	$P(\alpha)$



# Bilinear Pairings

Let  $G_1$  and  $G_2$  be two cyclic groups of prime order  $q$ . We write them multiplicatively.

Problem name	Input	Output
Decisional DH (DDH)	$g, g^a, g^b, g^c \in G_1$	TRUE iif $c = ab$
Computational DH (CDH)	$g, g^a, g^b \in G_1$	$g^{ab}$
Decisional Co-DH (co-DDH)	$g, g^b \in G_1$ $h, h^a \in G_2$	TRUE iif $a = b$
Computational Co-DH (co-CDH)	$g \in G_1$ $h, h^a \in G_2$	$g^a$

# Bilinear Pairings

## Definition 2.3 (Bilinear map).

Let  $G_T$  be an additional group s.t.  $|G_1| = |G_2| = |G_T|$ .

A **bilinear map** is a map  $e : G_1 \times G_2 \rightarrow G_T$  s.t.:

- Is bilinear:  $\forall u \in G_1, \forall v \in G_2, \forall a, b \in \mathbb{Z}$ ,

$$e(u^a, v^b) = e(u, v)^{ab}$$

- Is non-degenerate:  $e(g_1, g_2) \neq 1$ .

## Definition 2.4 (Gap problem).

A Gap co-Diffie-Hellman (co-GDH) group pair  $(G_1, G_2)$  is s.t. co-DDH is easy but co-CDH is hard. When there is an efficient isomorphism  $G_1 \cong G_2$  we say  $G_1$  is a Gap group (GDH).

# Secret Sharing

$\mathcal{P} := \{P_1, \dots, P_n\}$  set of participants.

## Definition 2.5 (Monotone Access Structure).

A **Monotone Access Structure**  $\Gamma$  is the set of all subsets of  $\mathcal{P}$  that can recover the secret, which is monotone increasing.

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

# Secret Sharing

## Definition 2.6.

A **perfect secret sharing scheme**, with respect to a monotone access structure  $\Gamma$  satisfies:

- If a subset  $A \in \Gamma$  of participants pool their shares, then can recover the secret.
- If a subset  $A \notin \Gamma$  of participants pool their shares, they can determine nothing about the secret.

## Definition 2.7.

An **anonymous secret sharing scheme** is a secret sharing scheme in which the secret can be reconstructed without the knowledge of which participants hold which shares.

# Shamir Secret Sharing

Shamir (1979). Goal: Share a secret  $s \in \mathbb{Z}_p$

- $s \in_R \mathbb{Z}_p$  the secret to be shared among  $\mathcal{P}$ .
- Set  $a_0 = s$  and choose  $a_1, \dots, a_{t-1} \in_R \mathbb{Z}_p$  with  $a_{t-1} \neq 0$   
Set  $P(x) = \sum_{i=0}^{t-1} a_i x^i$  polynomial of degree  $t-1$
- Choose  $\alpha_1, \dots, \alpha_n \in_R \mathbb{Z}_p^*$  all distinct.
- Each participant  $P_i \in \mathcal{P}$  is given the share  $(\alpha_i, y_i := P(\alpha_i))$
- The secret can be recovered with at least  $t$  shares with polynomial interpolation:

$$s = P(0) \leftarrow \sum_{j=1}^t y_{i_j} \prod_{k \in [t] \setminus \{j\}} \frac{-\alpha_{i_k}}{\alpha_{i_j} - \alpha_{i_k}}$$

# Digital Signatures

A Digital Signature Scheme consists of 3 algorithms:

- Key Generation:

- Input: Security parameter.
- Output: Pair  $(sk, pk)$  of secret and public keys.

- Sign:

- Input: Message  $m$ , secret key  $sk$ .
- Output: Signature  $\sigma$  on the message  $m$ .

- Verify:

- Input: Message  $m$ , signature  $\sigma$  on  $m$ , public key  $pk$ .
- Output: TRUE if the signature is valid. Otherwise FALSE.

# BLS Signature Scheme

Boneh, Lynn and Shacham (2001)

Let:

- $(G_1, G_2)$  a bilinear group pair of prime order  $p$
- $g$  a generator of  $G_1$
- $e : G_1 \times G_2 \rightarrow G_T$  a bilinear pairing.
- $H : \{0, 1\}^* \rightarrow G_1$  a full-domain hash function.

Key generation: Choose secret key  $x \in_R \mathbb{Z}_p$ . Set public key  $y := g_2^x$ .

Sign: Given message  $m \in \{0, 1\}^*$ , compute  $h := H(m) \in G_1$  and then compute the signature  $\sigma := h^x \in G_1$

Verify: Given public key  $y$ , message  $m$  and signature  $\sigma$ , compute  $h = H(m)$  and verify that  $e(\sigma, g_2) = e(h, y)$ .

# Group Signatures

Allow a member of the group to anonymously sign a message on behalf of the group Properties:

- Unforgeability
- Anonymity

Optional properties:

- Unlinkability
- Traceability



# Threshold Digital Signatures

## Definition 2.8.

*A  $(t, n)$ -threshold signature scheme is a signature scheme in which any set of  $t$  participants of the group is able to compute a signature on behalf of the group, and any subset of less than  $t$  participants is unable to compute a valid signature.*

# Example of Threshold Signature

Boldyreva (2003)

Setup Algorithm:

- $\mathcal{P} = \{P_i\}$  set of  $n$  participants.
- $G$  a Gap group of large prime order  $p > n$ , and  $g \in G$  a generator of the group.
- Choose  $sk \in_R \mathbb{Z}_p$  the secret key. Set  $pk = g^{sk}$  the public key.
- Set  $a_0 = sk$  and choose  $a_1, \dots, a_{t-1} \in_R \mathbb{Z}_p$  with  $a_{t-1} \neq 0$ .  
Set  $P(x) = \sum_{i=0}^{t-1} a_i x^i$ .
- Choose  $\alpha_1, \dots, \alpha_n \in_R \mathbb{Z}_p$  all distinct.
- Each participant  $P_i \in \mathcal{P}$  is given public key  $pk_i = \alpha_i$  and secret key  $sk_i = P(\alpha_i)$

## Sign Algorithm:

- $P = \{P_{i_1}, \dots, P_{i_t}\}$  set of  $t$  participants to sign message  $m$ .
- Each  $P_{i_j}$  computes partial signature  $\sigma_{i_j}(m) = H(m)^{sk_{i_j}}$
- Each  $P_{i_j}$  broadcasts the pair  $(pk_{i_j}, \sigma_{i_j}(m))$
- The signature  $\sigma$  on  $m$  is computed:

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

where  $\lambda_{i_j}^P := \prod_{k \in [t] \setminus \{j\}} \frac{-\alpha_{i_k}}{\alpha_{i_j} - \alpha_{i_k}}.$

## Verify Algorithm:

- $e : G \times G \rightarrow G_t$  bilinear pairing.
- $\sigma$  signature on a message  $m$ .
- $\sigma$  is valid  $\Leftrightarrow e(\sigma, g) = e(H(m), pk)$

# Anonymity

# Single Use: Anonymity

We now consider threshold signature schemes with:

- Non-traceability
- Linkability

We can avoid linkability by newly setting up the scheme after every signature.

# Solution with Anonymous Secret Sharing

- Signature scheme with secret key  $sk$ .
- Share the secret key among the set of participants using a  $(t, n)$ -anonymous threshold secret sharing scheme.
- To compute a signature: a set of  $t$  participants recover the secret key and compute the signature.

# Solution with Anonymized Threshold BLS Signatures

- Use BLS Threshold Signature Scheme: secret key  $sk \in_R \mathbb{Z}_p$  and random polynomial  $P(x)$  of degree  $t - 1$  s.t.  $P(0) = sk$
- To avoid linkability set up the scheme after a signature is computed
- To reach anonymity with respect to the dealer (who deals the shares) the participants choose random  $\alpha_i$  themselves and learn  $P(\alpha_i)$  using Oblivious Polynomial Evaluation.



# Multiple Use: Anonymity with Non-Linkability

We describe three solutions:

- Constant Size Signature Scheme
- Linkable Group Signature Scheme
- Anonymous Interactive Protocol

# Constant Size Anonymous Threshold Signature

Daza et al. (2009)

Setup Algorithm:

- Consider  $d$  distinct partitions of the set of participants  $\mathcal{P}$  into  $r$  parts:  $\mathcal{P}^i = \{\mathcal{P}_1^i, \dots, \mathcal{P}_r^i\}$ .
- For each partition  $\mathcal{P}^i, i \in [d]$  set up a  $(t, r)$ -Threshold BLS Signature scheme, and give same key pairs to all participants in the same  $\mathcal{P}_j^i$

## Sign Algorithm

- $\{P_{i_1}, \dots, P_{i_t}\}$  set of  $t$  participants to sign a message  $m$ .
- Signature on  $m$  over the  $i$ -th signature scheme is attempted.  
If succeeds, outputs  $(m, \sigma, i)$ .
- If signature fails (at least two participants have same secret key), a new signature over a distinct signature scheme is attempted.
- Eventually, the signature will succeed.

Verify Algorithm:

- Signature  $(m, \sigma, i)$ .
- Signature valid  $\Leftrightarrow e(\sigma, g) = e(H(m), pk_i)$

# Linkable Group Signature Scheme

Chen, Ng and Wang (2011)

Setup Algorithm:

- Participant generates a pair  $(sk, pk)$  of secret and public keys.
- Issuer gives the participant a credential that certifies the participant's public key as member of the group.

Sign and Verify Algorithms:

- Based on BLS Signature Scheme to verify credentials
- Based on Schnorr Signature Scheme to verify the signature

## Threshold Checking Algorithm:

- List of  $\ell$  valid signatures on  $m$ .
- Verify received signature  $\sigma$
- Check if already received same signature, or same signer signed twice.
- If not a duplicate, add  $\sigma$  to the list.
- When  $\ell = t$ , the threshold is reached, and the signature is the collection  $\{\sigma_i\}$  of  $t$  valid signatures on  $m$ .

# Anonymous Interactive Protocol

Set a Threshold BLS Signature Scheme

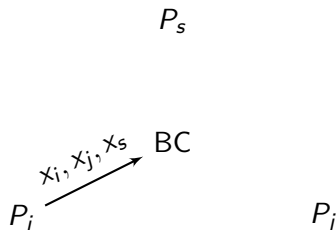
Each participant  $P_i$  owns pair  $(\alpha_i, P(\alpha_i))$  of public and secret keys

We propose an improvement on the signing algorithm s.t. the public key is not shared and cannot be obtained

# Interactive Protocol

- Interaction between  $P_i$ ,  $P_j$  and additional secure party  $P_s$ .
- Goal: given  $a \in G$ , compute  $a^{\frac{-\alpha_j}{\alpha_i - \alpha_j}}$  without sharing  $\alpha_i, \alpha_j$ .
- We will write:  $a^{\frac{-\alpha_j}{\alpha_i - \alpha_j}} \leftarrow \mathcal{B}(a, P_i, P_j)$





$$x_i, x_j, x_s \in_R \mathbb{Z}_p^*$$

$$\gamma_{i,0}, \gamma_{i,1}, \gamma_{i,2}, \gamma_{i,3}, \gamma_{i,4} \in_R \mathbb{Z}_p$$

$$g_i(x) \leftarrow \gamma_{i,1} \cdot x + \gamma_{i,0}$$

$$f_i(x) \leftarrow \gamma_{i,2} \cdot x + \alpha_i$$

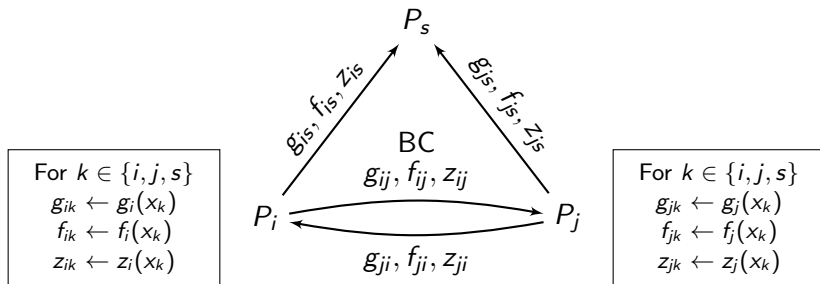
$$z_i(x) \leftarrow \gamma_{i,4} \cdot x + \gamma_{i,3}$$

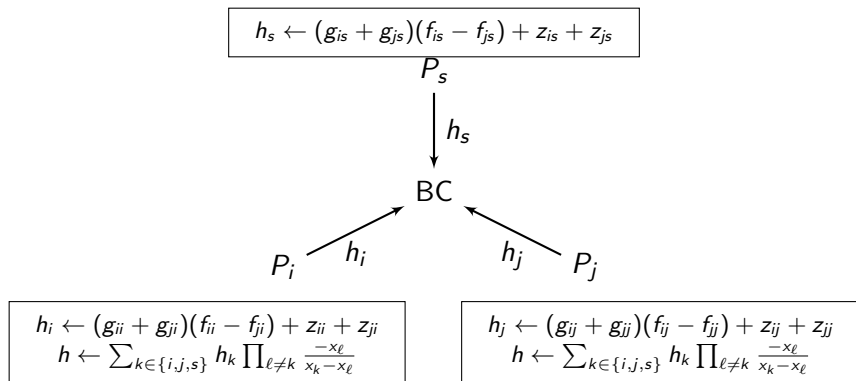
$$\gamma_{j,0}, \gamma_{j,1}, \gamma_{j,2}, \gamma_{j,3}, \gamma_{j,4} \in_R \mathbb{Z}_p$$

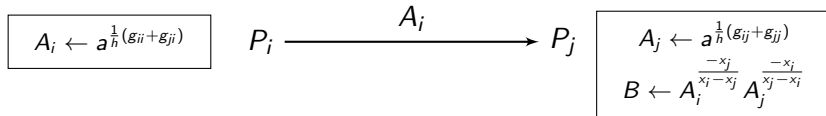
$$g_j(x) \leftarrow \gamma_{j,1} \cdot x + \gamma_{j,0}$$

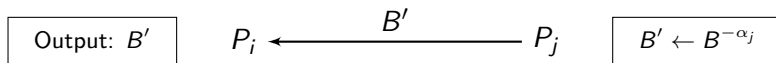
$$f_j(x) \leftarrow \gamma_{j,2} \cdot x + \alpha_j$$

$$z_j(x) \leftarrow \gamma_{j,4} \cdot x + \gamma_{j,3}$$









Where

$$B' = a^{\frac{-\alpha_j}{\alpha_i - \alpha_j}}$$

## Partial Signature:

- Let  $P = \{P_i, P_{j_1}, \dots, P_{j_{t-1}}\}$
- Let  $a_0 = H(m)^{s_i}$
- For  $k \in [t]$  compute

$$a_k \leftarrow \mathcal{B}(a_{k-1}, P_i, P_{j_k})$$

- $\sigma_i(m) = a_t = H(m)^{s_i \prod_{i \in [t-1]} \frac{-\alpha_{j_k}}{\alpha_{j_i} - \alpha_{j_k}}}$

## Signature:

$$\sigma(m) = \prod_{P_i \in P} \sigma_i(m)$$

## Unlinkability:

- Scheme remains unlinkable while participants honest but curious.
- If an adversary corrupts  $P_j, P_s$  she can interpolate  $\alpha_i$  from  $f_{ij}, f_{is}$
- To allow an adversary to corrupt up to  $\ell$  participants and still be unlinkable we can extend the algorithm:
  - $f_k, g_k$  polynomials of degree  $\ell + 1$
  - $2\ell + 3$  participants:  $P_i, P_j$ , and  $2\ell + 1$  secure parties.

## Complexity

- $P_i$  to compute  $\sigma_i$ :  $t - 1$  interactions.
- To compute  $\sigma$ :  $t(t - 1)$  interactions.



