Anonymous Threshold Signatures

Petar Hlad Colic

Universitat Politècnica de Catalunya

July 2018

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.
- Encription $\mathcal{E}$:
  - Input: Plaintext m
  - Output: Ciphertext $c = \mathcal{E}_{pk}(m)$
- Decryprtion $\mathcal{D}$:
  - Input: Ciphertext c
  - Output: Plaintext $m = \mathcal{D}_{sk}(c)$

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

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

Homomorphic PKE

Definition 1.1 (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 1.2.

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

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

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 multiplicativelly.

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)	$h, h^b \in G_1$ $g_2, g_2^a \in G_2$	TRUE iif $a = b$
Computational Co-DH (co-CDH)	$h \in G_1$ $g_2, g_2^a \in G_2$	h^a

Bilinear Pairings

Definition 1.3 (Bilinear map).

Let G_T be an additional group s.t. $|G_1| = |G_2| = |G_T|$.
A **bilinear map** *is a map $e : G_1 \times G_2 \to G_T$ 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 1.4 (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

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

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 $a_1, \ldots, a_{t-1} \in_R \mathbb{Z}_p$ with $a_{t-1} \neq 0$
  Set $P(x) = \sum_{i=0}^{t-1} a_i x^i$ polynomial of degree $t - 1$
- Choose $\alpha_1, \ldots, \alpha_n \in_R \mathbb{Z}_p^*$ 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:

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

Anonymous Secret Sharing

Definition 1.5.

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.

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 σ on the message m.
- Verify:
  - Input: Message m, signature σ 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
- $e : G_1 \times G_2 \to G_T$ a bilinear pairing.
- $H : \{0,1\}^* \to G_1$ 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 σ, 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 1.6.

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 $a_1, ..., a_{t-1} \in_R \mathbb{Z}_p$ with $a_{t-1} \neq 0$. Set $P(x) = \sum_{i=0}^{t-1} a_i x^i$.
- Choose $\alpha_1, ..., \alpha_n \in_R \mathbb{Z}_p$ 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:

- $P = \{P_{i_1}, ...P_{i_t}\}$ 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 σ 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:

- $e : G \times G \to G_t$ bilinear pairing.
- σ signature on a message m.
- σ 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 α_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: $\mathcal{P}^i = \{\mathcal{P}^i_1, ..., \mathcal{P}^i_r\}$.
- 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 $\mathcal{P}^i_j$

Sign Algorithm

- $\{P_{i_1}, ..., P_{i_t}\}$ set of t participants to sign a message m.
- Signature on m over the i-th signature scheme is attempted. If succeeds, outputs (m, σ, 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, σ, 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.
- The issuer gives the participant the credential (A, B, C) that authenticates the participant's public key as member of the group.