Fully Distributed Non-Interactive Adaptively-Secure Threshold Signature Scheme with Short Shares: Efficiency Considerations and Implementation

Benoît Libert[1,2] Marc Joye[3] Moti Yung[4] Fabrice Mouhartem[2,5]

NIST Threshold Cryptography Workshop 2019, March 11th

[1]CNRS, France

[2]École Normale Supérieure de Lyon, France

[3]OneSpan, Belgium

[4]Google Inc. and Columbia University, USA

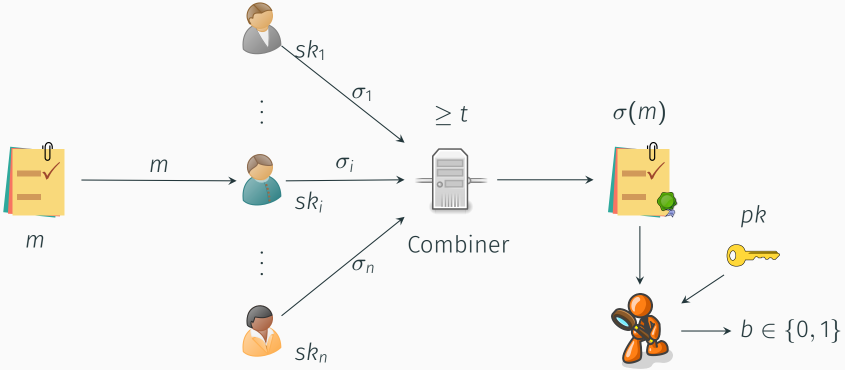
[5]Microsoft Research India

Motivation

Single points of failure is too risky:

- Surveillance by dedicated powerful adversaries (governments) on the Internet and its encryption and signing methods has been highlighted
- Attacks on certification authorities lead to fake certificates distributed over the Internet and destroy the "trust infrastructure"
  - e.g., DigiNotar was hacked and attacker produced a DigiNotar-signed "Google certificate" (September 2011)
  - Darkmatter's UAE security company, known for mass surveillance, requested to be a trusted CA (February 2019)
- Threshold cryptography (Desmedt-Frankel, Crypto'89 & Boyd, IMA'89) is a mechanism to deal with this by splitting keys among shareholders
  - Enhances the security of highly sensitive keys and the availability of systems

(t,n)-threshold signature scheme:



Constructing Threshold Cryptography

Two main design families for threshold cryptography:

- Drop-in replacement: e.g., threshold (EC)DSA, (ACNS'16), threshold RSA (Crypto'91)
  $\rightarrow$ Most of the proposed solutions
- Optimized threshold: achieve certain performance using the best secure scheme

Constructing Threshold Cryptography

Two main design families for threshold cryptography:

- Drop-in replacement: e.g., threshold (EC)DSA, (ACNS'16), threshold RSA (Crypto'91)
  $\rightarrow$ Most of the proposed solutions
- Optimized threshold: achieve certain performance using the best secure scheme

The second approach is more flexible (and allows proving adaptive security)

We optimize the following parameters simultaneously:

- Security
- Signature size
- Share size
- Communication

Static vs Adaptive Corruptions

- Static corruptions: adversary corrupts servers before seeing the *pk*
- First threshold signatures:
  - Desmedt-Frankel (Crypto'91): threshold RSA w/o robustness (heuristic)
  - De Santis *et al.* (STOC'94): provably secure, but large partial signatures
  - Gennaro *et al.* (Eurocrypt'96 & Crypto'96): threshold DSA & RSA signatures
  - Frankel *et al.* (FOCS'97 & Crypto'97): threshold RSA with interaction
  - ...

Static vs Adaptive Corruptions

- Static corruptions: adversary corrupts servers before seeing the *pk*
- First threshold signatures:
  - Desmedt-Frankel (Crypto'91): threshold RSA w/o robustness (heuristic)
  - De Santis *et al.* (STOC'94): provably secure, but large partial signatures
  - Gennaro *et al.* (Eurocrypt'96 & Crypto'96): threshold DSA & RSA signatures
  - Frankel *et al.* (FOCS'97 & Crypto'97): threshold RSA with interaction
  - ...
- Robust threshold signatures without interaction:
  - Shoup (Eurocrypt'00): practical threshold RSA signatures
  - Katz-Yung (Asiacrypt'02): threshold Rabin signatures
  - Boldyreva (PKC'03): short threshold signatures
  - Wee (Eurocrypt'11): generic constructions

Static vs Adaptive Corruptions

- Adaptive corruptions: adversary corrupts up to t servers *at any time*.
  - Canetti *et al.* (Crypto'99) and Frankel-MacKenzie-Yung (ESA'99, Asiacrypt'99): reliance on erasures
  - Jarecki-Lysyanskaya (Eurocrypt'00): no need for erasures, but much interaction at decryption
  - Lysyanskaya-Peikert (Asiacrypt'01): adaptively secure signatures with interaction.
  - Abe-Fehr (Crypto'04): adaptively secure UC-secure threshold signatures and encryption with interaction
  - Almansa-Damgaard-Nielsen (Eurocrypt'06): adaptively secure proactive RSA, but with interaction and $O(n)$ storage
  - Libert-Yung (ICALP'11): adaptively secure signatures without interaction, but using erasures and a trusted dealer

Threshold Signatures: Our Results (PODC'14, TCS'16)

- **Adaptively secure** threshold signatures have not been achieved with:
  - Non-interactivity
  - Robustness against malicious adversaries
  - Optimal resilience ($t = (n-1)/2$)
  - No erasures
  - Constant-size private key shares (regardless of t, n)
  - Distributed key generation (no trusted dealer)
- We give efficient candidates with one-round distributed key generation

Threshold Signatures: Our Results (PODC'14, TCS'16)

- **Adaptively secure** threshold signatures have not been achieved with:
  - Non-interactivity
  - Robustness against malicious adversaries
  - Optimal resilience ($t = (n-1)/2$)
  - No erasures
  - Constant-size private key shares (regardless of t, n)
  - Distributed key generation (no trusted dealer)

- We give efficient candidates with one-round distributed key generation

Theorem

In the random oracle model, constructions exist under standard assumptions

- Security under chosen-message attacks and adaptive corruptions:

  1. **Distributed key generation:** Challenger runs Dist-Keygen with $\mathcal{A}$

     $\mathcal{A}$ can corrupt players during Dist-Keygen and obtains PK, $\{SK_i\}_{i \in \mathcal{C}}$

Security of Non-interactive Threshold Signatures

- Security under chosen-message attacks and adaptive corruptions:

  1. **Distributed key generation:** Challenger runs Dist-Keygen with $\mathcal{A}$

     $\mathcal{A}$ can corrupt players during Dist-Keygen and obtains PK, $\{SK_i\}_{i \in \mathcal{C}}$

  2. **Query stage:** $\mathcal{A}$ makes adaptive queries

     - Corruption $i \in \{1, \ldots, n\}$: $\mathcal{A}$ receives SK_i
       and $\mathcal{C} := \mathcal{C} \cup \{i\}$ is updated

     - Signature (i, M): $\mathcal{A}$ receives $\sigma_i = Share\text{-}Sign(i, SK_i, M)$

Security of Non-interactive Threshold Signatures

- Security under chosen-message attacks and adaptive corruptions:

1. **Distributed key generation:** Challenger runs Dist-Keygen with $\mathcal{A}$

   $\mathcal{A}$ can corrupt players during Dist-Keygen and obtains PK, $\{SK_i\}_{i \in \mathcal{C}}$

2. **Query stage:** $\mathcal{A}$ makes adaptive queries

   - Corruption $i \in \{1, \ldots, n\}$: $\mathcal{A}$ receives SK_i
     and $\mathcal{C} := \mathcal{C} \cup \{i\}$ is updated
   - Signature (i, M): $\mathcal{A}$ receives $\sigma_i = \mathit{Share\text{-}Sign}(i, SK_i, M)$

3. **Output:** $\mathcal{A}$ outputs a pair $(M^\star, \sigma^\star)$ and wins if

   - Verify$(PK, M^\star, \sigma^\star) = 1$
   - $|\mathcal{V} \cup \mathcal{C}| \leq t$ where

$$\mathcal{V} := \{i \in \{1, \ldots, n\} \mid (i, M^\star) \text{ was queried by } \mathcal{A}\}$$

Construction in the random oracle model

Based on linearly **homomorphic structure-preserving** signatures (HSPS):
(Libert-Peters-Joye-Yung, Crypto'13)

- Messages are vectors $\vec{M} = (M_1, \ldots, M_N) \in \mathbb{G}^N$ in a **discrete-log-hard** group $\mathbb{G}$, for some $N \in \mathbb{N}$
- **Homomorphism:** given signatures $\{\sigma_i\}_{i=1}^{\ell}$ on vectors $\{\vec{M}_i\}_{i=1}^{\ell}$, anyone can compute a signature $\sigma = \prod_{i=1}^{\ell} \sigma_i^{\omega_i}$ on a linear combination $\vec{M} = \prod_{i=1}^{\ell} \vec{M}_i^{\omega_i}$
- **Security:** deriving a signature for $\vec{M} \notin \mathrm{span}(\vec{M}_1, \ldots, \vec{M}_\ell)$ is infeasible
- For $N > 1$, deciding if $\vec{M}_1, \ldots, \vec{M}_{\ell+1} \in \mathbb{G}^N$ are linearly independent is hard

Definition (K-linear assumption)

given vectors $\vec{g}_1, \ldots, \vec{g}_{K+1} \in_R \mathbb{G}^{K+1}$, no PPT algorithm can decide if $\dim(\langle \vec{g}_1, \ldots, \vec{g}_{k+1} \rangle) = K$ or $K+1$

- Let $\mathbf{\Pi} = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Derive})$ be a HSPS scheme
- Signature scheme based on the K-linear assumption
  - $\mathsf{Keygen}(1^\lambda)$: runs $(pk, sk) \leftarrow \mathbf{\Pi}.\mathsf{Keygen}(1^\lambda, K+1)$ and chooses a hash function $H : \{0,1\}^* \rightarrow \mathbb{G}^{K+1}$
  - $\mathsf{Sign}(sk, M)$: computes $(H_1, \ldots, H_{K+1}) = H(M) \in \mathbb{G}^{K+1}$ and outputs
    $$\sigma \leftarrow \mathbf{\Pi}.\mathsf{Sign}(sk, (H_1, \ldots, H_{K+1}))$$
  - $\mathsf{Verify}(pk, M, \sigma)$: computes $(H_1, \ldots, H_{K+1}) = H(M) \in \mathbb{G}^{K+1}$ and returns 1 if and only if $\mathbf{\Pi}.\mathsf{Verify}(pk, (H_1, \ldots, H_{K+1}), \sigma) = 1$

Theorem

In the ROM, the scheme is secure against chosen-message attacks if the K-linear assumption holds in $\mathbb{G}$.

Distributing the system using specific properties of our HSPS:

- **Key-homomorphism:** For any $\vec{M}$, given $\sigma_1 \leftarrow \mathsf{Sign}(sk_1, \vec{M})$ and $\sigma_2 \leftarrow \mathsf{Sign}(sk_2, \vec{M})$, anyone can compute $\sigma \leftarrow \mathsf{Sign}(sk_1 + sk_2, \vec{M})$

  $\Rightarrow$ Convenient for non-interactive threshold signing
  (reconstruction via interpolation in the exponent)

Construction (cont.)

Distributing the system using specific properties of our HSPS:

- **Key-homomorphism:** For any $\vec{M}$, given $\sigma_1 \leftarrow \mathsf{Sign}(sk_1, \vec{M})$ and $\sigma_2 \leftarrow \mathsf{Sign}(sk_2, \vec{M})$, anyone can compute $\sigma \leftarrow \mathsf{Sign}(sk_1 + sk_2, \vec{M})$

  $\Rightarrow$ Convenient for non-interactive threshold signing
  (reconstruction via interpolation in the exponent)

- In the security proof, the private key is known at any time

  $\Rightarrow$ Allows handling *adaptive* corruption queries.

Construction (cont.)

- Key is generated using Pedersen's distributed key generation (DKG) protocol (Eurocrypt'91)
  - Only one round without faulty players
  - ...but does **not** guarantee **uniform** keys, even for static adversaries (Gennaro-Jarecki-Krawczyk-Rabin, Eurocrypt'99)

  $\Rightarrow$ Reductions from a centralized scheme are impossible

- It is sometimes possible to prove security using direct proofs (Gennaro-Jarecki-Krawczyk-Rabin, CT-RSA'03)
  - This approach is more suitable for optimized constructions

Construction (cont.)

- Based on bilinear maps (a.k.a. pairings)

$$e : \mathbb{G} \times \hat{\mathbb{G}} \rightarrow \mathbb{G}_T$$

  such that

$$e(g^a, \hat{h}^b) = e(g^b, \hat{h}^a) = e(g, \hat{h})^{ab} \quad \forall g \in \mathbb{G},\ \hat{h} \in \hat{\mathbb{G}},\ a, b \in \mathbb{Z}$$

- We assume the hardness of the **Decision Diffie-Hellman** (DDH problem) in $\mathbb{G}$ and $\hat{\mathbb{G}}$:

Definition (DDH Problem)

In a cyclic group $G = \langle g \rangle$ of order p, given $(g, g^a, g^b, T) \in G^4$, decide whether $T = g^{ab}$ or $T \in_R G$

(Coincides with the K-linear assumption for $K = 1$)

Outline of the Scheme

- Public key is

$$PK = \left(\hat{g},\ \hat{h},\ \{\hat{g}_k = \hat{g}^{a_k} \cdot \hat{h}^{b_k}\}_{k=1}^{2}\right) \in \hat{\mathbb{G}}^4$$

and $SK = \{(a_k, b_k)\}_{k=1}^2$ is shared as $SK_i = \{(A_k(i), B_k(i))\}_{k=1}^2$ using

$$A_k[X] = a_{k0} + a_{k1}X + \cdots + a_{kt}X^t, \qquad B_k[X] = b_{k0} + b_{k1}X + \cdots + b_{kt}X^t$$

- Public key is

$$PK = (\hat{g},\ \hat{h},\ \{\hat{g}_k = \hat{g}^{a_k} \cdot \hat{h}^{b_k}\}_{k=1}^{2}) \in \hat{\mathbb{G}}^4$$

and $SK = \{(a_k, b_k)\}_{k=1}^2$ is shared as $SK_i = \{(A_k(i), B_k(i))\}_{k=1}^2$ using

$$A_k[X] = a_{k0} + a_{k1}X + \cdots + a_{kt}X^t, \qquad B_k[X] = b_{k0} + b_{k1}X + \cdots + b_{kt}X^t$$

- Player i computes $(H_1, H_2) = H(M) \in \mathbb{G}^2$ and

$$(z_i, r_i) = \Big(\prod_{k=1}^{2} H_k^{A_k(i)},\ \prod_{k=1}^{2} H_i^{B_k(i)}\Big).$$

- Public key is

$$PK = (\hat{g},\ \hat{h},\ \{\hat{g}_k = \hat{g}^{a_k} \cdot \hat{h}^{b_k}\}_{k=1}^{2}) \in \hat{\mathbb{G}}^4$$

and $SK = \{(a_k, b_k)\}_{k=1}^{2}$ is shared as $SK_i = \{(A_k(i), B_k(i))\}_{k=1}^{2}$ using

$$A_k[X] = a_{k0} + a_{k1}X + \cdots + a_{kt}X^t, \qquad B_k[X] = b_{k0} + b_{k1}X + \cdots + b_{kt}X^t$$

- Player i computes $(H_1, H_2) = H(M) \in \mathbb{G}^2$ and

$$(z_i, r_i) = \Big(\prod_{k=1}^{2} H_k^{A_k(i)},\ \prod_{k=1}^{2} H_i^{B_k(i)}\Big).$$

For any $(t+1)$-set $S \subset \{1, \ldots, n\}$, partial signatures $\{(z_i, r_i)\}_{i \in S}$ yield

$$(z, r) = \Big(\prod_{i \in S} z_i^{\Delta_{i,S}(0)},\ \prod_{i \in S} r_i^{\Delta_{i,S}(0)}\Big),$$

such that $e(z, \hat{g}) \cdot e(r, \hat{h}) \cdot \prod_{k=1}^{2} e(H_k, \hat{g}_k) = 1_{\mathbb{G}_T}$

Theorem

In the ROM, the fully distributed scheme is adaptively secure (under chosen-message attacks) if the DDH problem is hard in $\mathbb{G}$ and $\hat{\mathbb{G}}$

Proof idea:

- $PK = (\hat{g}, \hat{h}, \{\hat{g}^{a_k}\hat{h}^{b_k}\}_{k=1}^{2})$ reveals limited information about $\{(a_k, b_k)\}_{k=1}^{2}$
- For any message M, two distinct signatures allow breaking DDH in $\hat{\mathbb{G}}$
- **Strategy:** get the adversary to produce a different forgery $\sigma^\star$ than the reduction's for $M^\star$
- **Problem:** PK is not uniform
- For each $k \in \{1, 2\}$, $(a_k, b_k) = (a_{k,\mathcal{G}} + a_{k,\mathcal{Q}\setminus\mathcal{G}}, b_{k,\mathcal{G}} + b_{k,\mathcal{Q}\setminus\mathcal{G}})$
- Key homomorphism allows turning a forgery for the private key $\{(a_k, b_k)\}_{k=1}^{2}$ into a forgery for the key $\{(a_{k,\mathcal{G}}, b_{k,\mathcal{G}})\}_{k=1}^{2}$

Security

Theorem

In the ROM, the fully distributed scheme is adaptively secure (under chosen-message attacks) if the DDH problem is hard in $\mathbb{G}$ and $\hat{\mathbb{G}}$

Proof idea (cont.):

- **Other problem:** make sure that signing queries do not leak too much information on $\{(a_{k,\mathcal{G}}, b_{k,\mathcal{G}})\}_{k=1}^{2}$
- Program $H : \{0,1\}^* \to \mathbb{G}^2$ so that
  - $H(M^\star) \in_R \mathbb{G}^2$ for the forgery message $M^\star$
  - $H(M) \in \mathbb{G}^2$ lives in a 1-dimensional subspace of $\mathbb{G}^2$ for each $M \neq M^\star$

  Change not noticeable if DDH is hard in $\mathbb{G}$
- With probability $\Theta(1/q)$, the reduction gets two distinct signatures for a uniform key $\{(a_{k,\mathcal{G}}, b_{k,\mathcal{G}})\}_{k=1}^{2}$

Performances

Algorithm	Dist-KeyGen	Share-Sign	Combine	Verify
(1, 2)	26	2	23	11
(5,11)	787	13	69	11
(11,20)	4 371	22	137	12
(26,51)	202 763	112	493	13

Table 1: PoC implementation results for (t, n)-threshold signatures in ms

Remarks on the implementation:

- It is a proof of concept implementation in `C++` and is not optimized
- It is sequential and does not capture parallel computations
- Uses a wrapper on the `Relic` toolkit for pairing computations

Source code available: `https://gitlab.inria.fr/fmouhart/threshold-signature`

Summary

- Our results: an optimized threshold construction from pairings
  - First adaptively secure non-interactive threshold signatures with
    - Robustness, $O(1)$-size private key shares, no erasures
    - One-round distributed key generation
    - Short signatures (i.e., 512 bits at the 128-bit security level) in the ROM
  - The construction can be made proactive (Ostrovsky-Yung, PODC'91)
- Open problems:
  - Construction in the standard model with short public parameters
  - Constructions based on the hardness of search (rather than decision) problems (e.g., RSA or computational Diffie-Hellman)

Thank you for your attention.