Round Optimal Secure Multisignature Schemes from Lattice with Public Key Aggregation and Signature Compression

Meenakshi Kansal and Ratna Dutta

Department of Mathematics
Indian Institute of Technology Kharagpur
INDIA

Multisignature

Unlike ordinary digital signature schemes, multisignatures are aimed to achieve the following objectives.

(i) A group of users will collaboratively produce the signature on the same message.

(ii) The size of the multisignature is asymptotically same as the size of a single signature.

(iii) The public keys should also be aggregated into a single public key where the size of the aggregated public key is asymptotically same as that of a single public key.

Security. No one should be able to approve the document on behalf of any of the N signers.

Unforgeability: Even if a single signer say U_j is honest among U_i's then any set of signers $\{U_i\}_{i=1}^{l}$ containing U_j should not produce a valid multisignature.

Plain Public Key Model. In this model, the users do not need to prove the knowledge or possession of their secret key.

Application. Storage and bandwidth costs are subject to minimization like blockchain.

Multisignature: History

1983 Itakura and Nakamura: Introduced multisignature.

2016 Bansarkhani and Sturm: First lattice based multisignature.

2018 Boneh, Drijvers and Neven: First compact multisignature and short accountable subgroup multisignature using pairings.

2020 Kansal and Dutta: First lattice based multisignature achieving both public key aggregation and signature compression. First lattice based accountable subgroup multisignature.

Table : Comparative summary of multisignature resistant to rogue key attack and secure in the ROM

MS	Communication		R_s	Storage		Computation	
	|apk|	|msig|		|pk|	|sk|	sign	verify
[1]	$\|\mathbb{G}_2\|$	$\|\mathbb{G}_1\|$	1	$\|\mathbb{G}_2\|$	$\|\mathbb{Z}_q\|$	1E	2P
[2]	$\|\mathbb{G}\|$	$2\|\mathbb{G}\| + 3\|\mathbb{Z}_q\|$	2	$\|\mathbb{G}\| + 2\|\mathbb{Z}_q\|$	$\|\mathbb{Z}_q\|$	5E	6E
[3]	$\|\mathbb{G}\|$	$2\|\mathbb{G}\|$	1	$\|\mathbb{G}\| + 2\|\mathbb{Z}_q\|$	$\mathcal{O}(l^2)$	4E	3P+1E
[4]	$\|\mathbb{G}\|$	$\|\mathbb{G}\| + \|\mathbb{Z}_q\|$	3	$\|\mathbb{G}\|$	$\|\mathbb{Z}_q\|$	2E	1E
[5]	–	$\mathcal{O}(\lambda)$	3	$\mathcal{O}(\lambda)$	$\mathcal{O}(\lambda)$	2PM	$(N+1)$PM
Ours	$\mathcal{O}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	1	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	2MM	2MM

[1] Dan Boneh, Manu Drijvers, and Gregory Neven. Compact multisignatures for smaller blockchains. In International Conference on the Theory and Application of Cryptology and Information Security, pages 435464. Springer, 2018.

[2] Manu Drijvers, Kasra Edalatnejad, Bryan Ford, Eike Kiltz, Julian Loss, Gregory Neven, and Igors Stepanovs. On the security of two-round multisignatures. In On the Security of Two-Round Multi-Signatures, page 0. IEEE, 2019.

[3] Manu Drijvers, Sergey Gorbunov, Gregory Neven, and Hoeteck Wee. Pixel: Multisignatures for consensus.

[4] Gregory Maxwell, Andrew Poelstra, Yannick Seurin, and Pieter Wuille. Simple schnorr multisignatures with applications to bitcoin. Designs, Codes and Cryptography, 87(9):21392164, 2019.

[5] Rachid El Bansarkhani and Jan Sturm. An efficient lattice-based multisignature scheme with applications to bitcoins. In International Conference on Cryptology and Network Security, pages 140155. Springer, 2016.

Table : Comparative summary of accountable subgroup multisignature resistant to rogue key attack and secure in the ROM

ASM	Communication		Storage			Computation	
	$\|\text{apk}\|$	$\|\text{msig}\|$	$\|\text{pk}\|$	$\|\text{sk}\|$	$\|\text{mk}\|$	sign	verify
[1]	$\|\mathbb{G}_2\|$	$\|\mathbb{G}_1\| + \|\mathbb{G}_2\|$	$\|\mathbb{G}_2\|$	$\|\mathbb{Z}_q\|$	$\|\mathbb{Z}_q\|$	1E	3P
Ours	$\mathcal{O}(\lambda^2)$	$\mathcal{O}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\mathcal{O}(\lambda^2)$	1MM	$(2 + \|S\|)$MM

$|\text{apk}|$: size of the aggregated public key, $|\text{msig}|$: size of the compressed signature, $|\text{pk}|$: size of a public key, $|\text{sk}|$: size of a secret key, $|\text{mk}|$: size of group membership key, $\mathbb{G}$, $\mathbb{G}_1$, $\mathbb{G}_2$ are groups of prime order q, $|\mathbb{G}|$: bit size of an element of the group $\mathbb{G}$, λ: security parameter, R_s: number of rounds in the signature generation algorithm, E: number of exponentiations, P: number of pairings, N: number of signers, $|S|$: size of the subgroup S, PM: number of polynomial multiplications, MM: number of matrix multiplications.

[1] Dan Boneh, Manu Drijvers, and Gregory Neven. Compact multisignatures for smaller blockchains. In International Conference on the Theory and Application of Cryptology and Information Security, pages 435464. Springer, 2018.

The MS

$\mathbf{MS.pg}(1^\lambda) \to (\mathcal{Y} = (n, q, m, \sigma, H_0, H_1, H_2, \mathbf{A}))$.

- n of size $\mathcal{O}(\lambda)$,
- q of size $\mathcal{O}(n^3)$,
- $m \geq 2n\lceil \log q \rceil$,
- σ of size $\Omega(\sqrt{n \log q} \log n)$,
- $H_0 : \{0,1\}^* \to \mathbb{Z}_q^{m \times n}$,
- $H_1 : \{0,1\}^* \to D_{\mathbb{Z}_q, \sigma}^{m \times n}$,
- $H_2 : \{0,1\}^* \to D_{\mathbb{Z}_q, \sigma}^{n \times n}$,
- $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$.

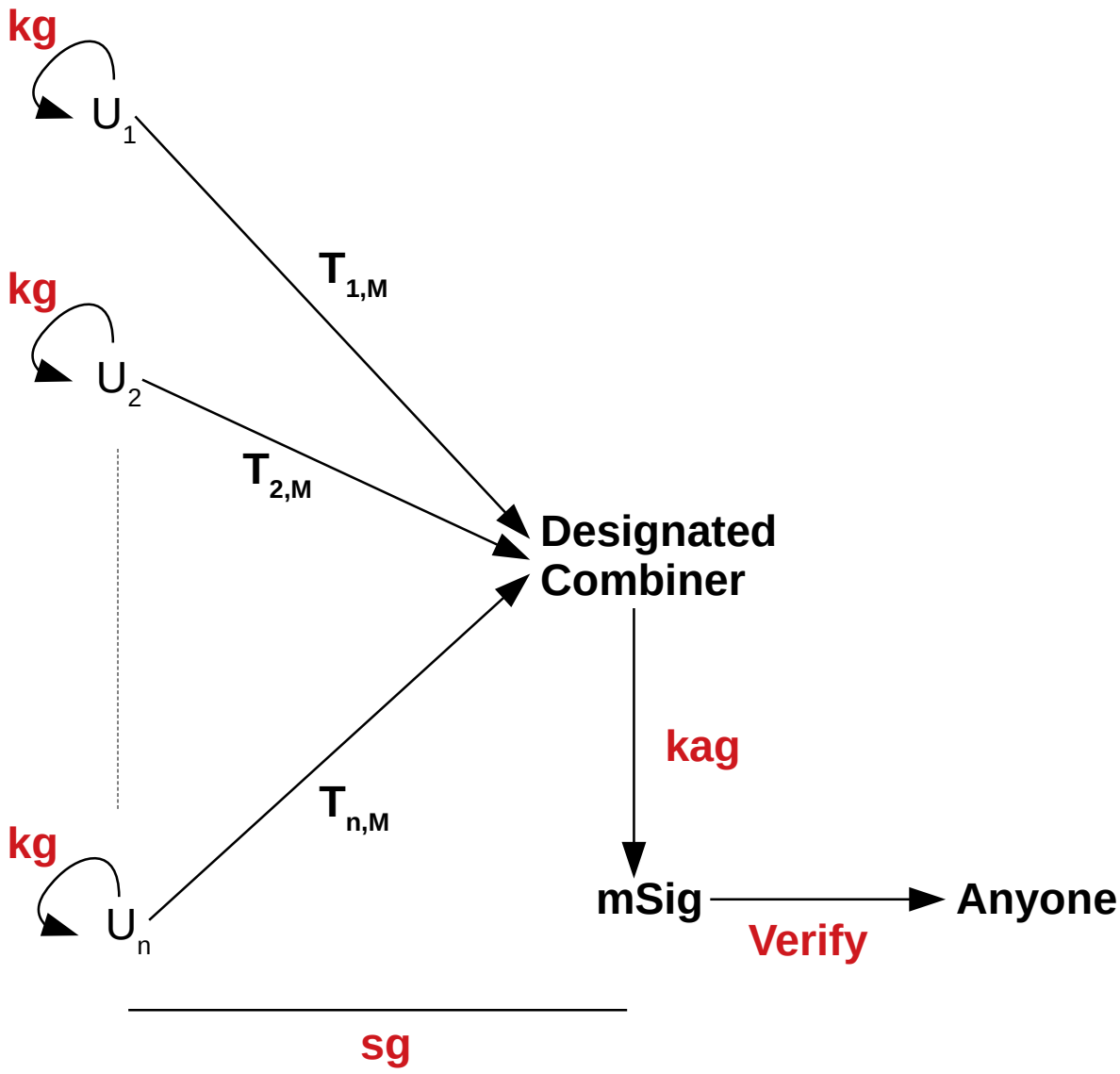
$\mathbf{MS.kg}(\mathcal{Y}, i) \rightarrow (\mathsf{pk}_i, \mathsf{sk}_i)$.

- Chooses $\mathbf{V}_i \in D^{m \times m}_{\mathbb{Z}_q, \sigma}$.
- Computes $\mathbf{Y}_i = \mathbf{A} \cdot \mathbf{V}_i \bmod q \in \mathbb{Z}_q^{n \times m}$.
- Sets $\mathsf{pk}_i = \mathbf{Y}_i \in \mathbb{Z}_q^{n \times m}$ and $\mathsf{sk}_i = \mathbf{V}_i \in D^{m \times m}_{\mathbb{Z}_q, \sigma}$.

$\mathbf{MS.kag}(\mathcal{Y}, \mathcal{PK}) \rightarrow \mathsf{pkag}_{\mathcal{PK}}$.

- $\mathcal{PK} = \{\mathsf{pk}_{i_1}, \mathsf{pk}_{i_2}, \ldots, \mathsf{pk}_{i_l}\}$.
- $I_{\mathcal{PK}} = \{i_1, i_2, \ldots, i_l\}$.
- computes $\mathsf{pkag}_{\mathcal{PK}} = \sum_{i \in I_{\mathcal{PK}}} \mathsf{pk}_i \cdot H_1(\mathsf{pk}_i, \mathcal{PK}) \in \mathbb{Z}_q^{n \times n}$.

Figure : Multisignature Outlook



$\mathbf{MS.sg}(\mathcal{Y}, \mathcal{PK}, \mathcal{SK}, M) \rightarrow \mathsf{msig}_{\mathcal{PK},M}$. Each signer $i \in I_{\mathcal{PK}}$ does the following.

- Generates

$$\mathbf{T}_{i,M} = H_0(M, \mathcal{PK}) + \mathsf{sk}_i \cdot H_1(\mathsf{pk}_i, \mathcal{PK}) \cdot H_2(M).$$

- Sends $\mathbf{T}_{i,M}$ to the designated signer.

The designated signer does the following.

- Verify

$$||\mathbf{T}_{i,M}|| \leq ||H_0(M, \mathcal{PK})|| + \sigma^3 m\sqrt{n},$$

$$\mathbf{A} \cdot \mathbf{T}_{i,M} = \mathbf{A} \cdot H_0(M, \mathcal{PK}) + \mathbf{Y}_i \cdot H_1(\mathsf{pk}_i, \mathcal{PK}) \cdot H_2(M).$$

- If the verification fails, return $\perp$.
- Otherwise, issue the multisignature $\mathsf{msig}_{\mathcal{PK},M}$ where

$$\mathsf{msig}_{\mathcal{PK},M} = \left(\mathbf{T}_M, \mathsf{pkag}_{\mathcal{PK}}, I_{\mathcal{PK}}, M\right),$$

$$\mathbf{T}_M = \sum_{i \in I_{\mathcal{PK}}} \mathbf{T}_{i,M} \bmod q$$

$\mathbf{MS.vrf}(\mathcal{Y}, \mathsf{msig}_{\mathcal{PK},M}) \rightarrow (0 \text{ or } 1)$.

- Verify

$$\mathbf{A} \cdot \mathbf{T}_M = \mathbf{A} \cdot |I_{\mathcal{PK}}| \cdot H_0(M, \mathcal{PK}) + \mathsf{pkag}_{\mathcal{PK}} \cdot H_2(M),$$

$$||\mathbf{T}_M|| \leq |I_{\mathcal{PK}}| \cdot (||H_0(M, \mathcal{PK})|| + \sigma^3 m\sqrt{n}).$$

Theorem 1

suppose that there exists a forger $\mathcal{F}$ running in time $t_{\mathcal{F}}$ can break the security under unforgeability of our scheme MS *with non-negligible advantage $\epsilon_{\mathcal{F}}$ making q_s signature queries and q_H hash queries. Then there exists an algorithm $\mathcal{S}$ running in time $(t_{\mathcal{F}} + t_{q_H} + t_{q_s} + t_{extra}) \cdot 8q_H^2 \cdot \epsilon_{\mathcal{F}} \cdot \log(8q_H/\epsilon_{\mathcal{F}})$, that for a given $\mathbf{P} \in \mathbb{Z}_q^{n \times m}$ finds a nonzero $\mathbf{V} \in \mathbb{Z}_q^{m \times m}$ satisfying $||\mathbf{V}|| \leq \sigma\sqrt{m}$ and $\mathbf{P} \cdot \mathbf{V} = \mathbf{0} \bmod q$ with non negligible advantage $\frac{\epsilon_{\mathcal{F}}}{8q_H}$. Here $m \geq 2n\lceil \log q \rceil$, σ is of size $\Omega(\sqrt{n \log q} \log n)$, q is of size $\mathcal{O}(n^3)$, t_{q_H}, t_{q_s} respectively denote the time taken to answer hash and signature queries and t_{extra} is extra time taken by the algorithm $\mathcal{S}$.*

Accountable Subgroup Multisignature

It enables a subset S of a set of potential signers G to jointly produce a multisignature on a given message such that it satisfies *flexibility* and *accountability*.

- Flexibility means that the verification is upto the verifier.

  For instance, consider a case when a company X signs a contract of a company Y. Suppose a subset S of X containing chief operating officer, chief financial officer and chief marketing officer sign the contract and sends the signature to Y. If Y prefers to have the signature of the chief executive officer then Y may reject the signature.
- Accountability refers to the fact that the set S is known to the verifier.

The ASM

$\mathsf{ASM.pg}(1^\lambda) \to \mathcal{Y} = (n, q, m, \sigma, H_0, H_1, H_2, H_3, \mathbf{A})$.

- n of size $\mathcal{O}(\lambda)$, q of size $\mathcal{O}(n^3)$, $m \geq 2n\lceil \log q \rceil$, σ of size $\Omega(\sqrt{n \log q} \log n)$,
- $H_0 : \{0,1\}^* \to \mathbb{Z}_q^{m \times n}$, $H_1 : \{0,1\}^* \to D_{\mathbb{Z}_q, \sigma}^{m \times n}$, $H_2 : \{0,1\}^* \to \mathbb{Z}_q^{n \times n}$, $H_3 : \{0,1\}^* \to D_{\mathbb{Z}_q, \sigma}^{n \times n}$ where $D_{\mathbb{Z}_q, \sigma}^{k \times l} = \{\mathbf{M} \in \mathbb{Z}_q^{k \times l} : ||\mathbf{M}|| \leq \sigma\sqrt{k}\}$,
- $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$.

$\mathbf{ASM.kg}(\mathcal{Y}, i) \to (\mathsf{pk}_i, \mathsf{sk}_i)$. Each user i does the following.

- Choose $\mathbf{V}_i \in D_{\mathbb{Z}_q, \sigma}^{m \times m}$ and compute $\mathbf{Y}_i = \mathbf{A} \cdot \mathbf{V}_i \bmod q \in \mathbb{Z}_q^{n \times m}$.
- Set $\mathsf{pk}_i = \mathbf{Y}_i \in \mathbb{Z}_q^{n \times m}$ and $\mathsf{sk}_i = \mathbf{V}_i \in D_{\mathbb{Z}_q, \sigma}^{m \times m}$.

$\mathbf{ASM.kag}(\mathcal{Y}, \mathcal{PK}) \to \mathsf{pkag}_{\mathcal{PK}}$.

- Outputs the aggregated public key

$$\mathsf{pkag}_{\mathcal{PK}} = \sum_{i \in I_{\mathcal{PK}}} \mathsf{pk}_i \cdot H_1(\mathsf{pk}_i, \mathcal{PK}) \in \mathbb{Z}_q^{n \times n}.$$

$\mathbf{ASM.gmk}(\mathcal{Y}, \mathcal{PK}, \mathcal{SK}_{\mathcal{PK}}) \to \mathsf{mk}_{i,\mathcal{PK}}$. Each user $i \in I_{\mathcal{PK}}$ does the following.

- Generate $\mathsf{pkag}_{\mathcal{PK}} \leftarrow \mathsf{ASM.kag}(\mathcal{Y}, \mathcal{PK})$.
- Compute $\mathbf{M}_{j,i} = H_2(\mathsf{pkag}_{\mathcal{PK}}, j) + \mathsf{sk}_i \cdot H_1(\mathsf{pk}_i, \mathcal{PK}) \cdot H_3(j)$ for all $j \in I_{\mathcal{PK}}$.
- Send $\mathbf{M}_{j,i}$ to signer j with $||\mathbf{M}_{j,i}|| \leq ||H_2(\mathsf{pkag}_{\mathcal{PK}}, j)|| + \sigma^3 m\sqrt{n}$.

On receiving $\mathbf{M}_{i,j} \setminus \{i\}$ from all signers $j \in I_{\mathcal{PK}}$, the i-th signer verifies

- $\mathbf{A} \cdot \mathbf{M}_{i,j} = \mathbf{A} \cdot H_2(\mathsf{pkag}_{\mathcal{PK}}, i) + \mathsf{pk}_j \cdot H_1(\mathsf{pk}_j, \mathcal{PK}) \cdot H_3(i)$.
- $||\mathbf{M}_{i,j}|| \leq ||H_2(\mathsf{pkag}_{\mathcal{PK}}, i)|| + \sigma^3 m\sqrt{n}$.
- If the verification fails, it returns $\perp$.
- Otherwise, it computes the group membership key

$$\mathsf{mk}_{i,\mathcal{PK}} = \sum_{j \in I_{\mathcal{PK}}} \mathbf{M}_{i,j} = \sum_{j \in I_{\mathcal{PK}}} \left[H_2(\mathsf{pkag}_{\mathcal{PK}}, i) + \mathsf{sk}_j \cdot H_1(\mathsf{pk}_j, \mathcal{PK}) \cdot H_3(i) \right]$$

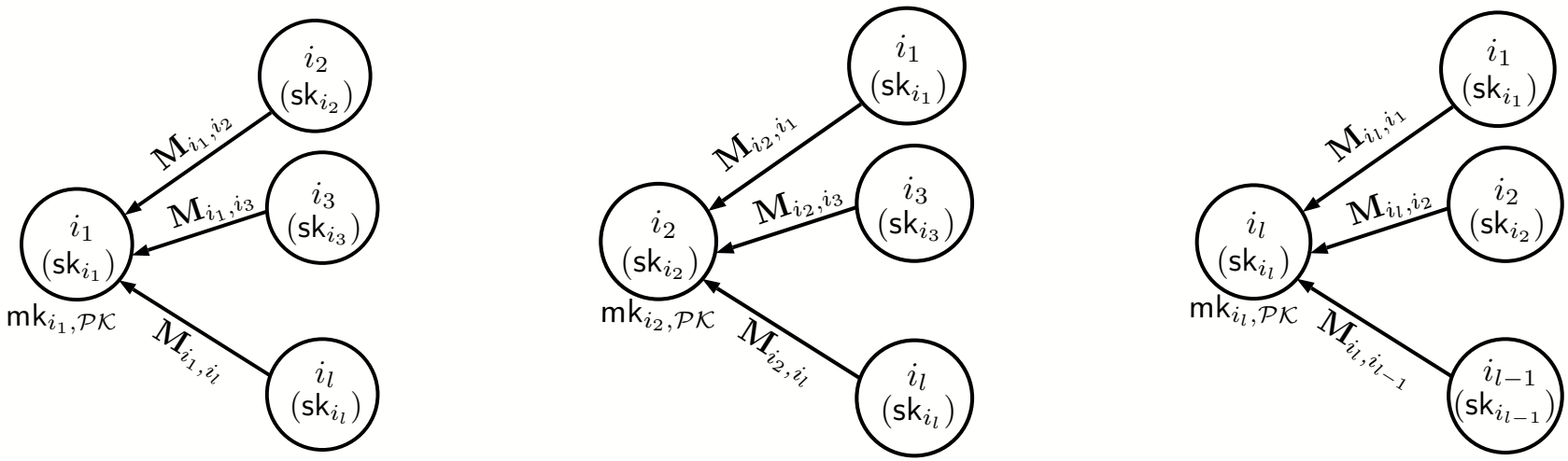


Figure : Group membership key generation of an user $i_j \in I_{\mathcal{PK}}$ where $\mathbf{M}_{i,j} = H_2(\mathsf{pkag}_{\mathcal{PK}}, i) + \mathsf{sk}_j \cdot H_1(\mathsf{pk}_j, \mathcal{PK}) \cdot H_3(i)$ for $j \in \mathcal{PK}$ and $\mathsf{mk}_{i,\mathcal{PK}} = \sum_{j \in I_{\mathcal{PK}}} \mathbf{M}_{i,j}$

$\mathbf{ASM.sg}(\mathcal{Y}, L, \mathcal{PK}, \mathcal{SK}_L, \mathcal{G}_L, M) \rightarrow \mathsf{accmsig}_{\mathcal{PK},L,M}$. Each signer $i \in I_L$ performs the following steps.

- Generate $\mathsf{pkag}_{\mathcal{PK}} \leftarrow \mathsf{ASM.kag}(\mathcal{Y}, \mathcal{PK})$ where $\mathsf{pkag}_{\mathcal{PK}} = \sum_{i \in I_{\mathcal{PK}}} \mathsf{pk}_i \cdot H_1(\mathsf{pk}_i, \mathcal{PK})$.
- Compute $\mathbf{T}_{i,M} = \mathsf{sk}_i \cdot H_0(\mathsf{pkag}_{\mathcal{PK}}, M) + \mathsf{mk}_{i,\mathcal{PK}}$ with $||\mathbf{T}_{i,M}|| \leq \sigma\sqrt{m} \cdot H_0(\mathsf{pkag}_{\mathcal{PK}}, M) + |I_{\mathcal{PK}}| \cdot ||H_2(\mathsf{pkag}_{\mathcal{PK}}, i)|| + |I_{\mathcal{PK}}| \cdot \sigma^3 m\sqrt{n}$.
- Send $\mathbf{T}_{i,M}$ to the designated signer.
- The designated signer verifies

$$\mathbf{A} \cdot \mathbf{T}_{i,M} = \mathsf{pk}_i \cdot H_0(\mathsf{pkag}_{\mathcal{PK}}, M) + \sum_{j \in I_{\mathcal{PK}}} \left[\mathbf{A} \cdot H_2(\mathsf{pkag}_{\mathcal{PK}}, i) + \mathsf{pk}_j \cdot H_1(\mathsf{pk}_j, \mathcal{PK}) \cdot H_3(i)\right],$$

$$||\mathbf{T}_{i,M}|| \leq \sigma\sqrt{m} \cdot H_0(\mathsf{pkag}_{\mathcal{PK}}, M) + |I_{\mathcal{PK}}| \cdot ||H_2(\mathsf{pkag}_{\mathcal{PK}}, i)|| + |I_{\mathcal{PK}}| \cdot \sigma^3 m\sqrt{n}.$$

- If it does not pass the verification, it aborts and returns $\perp$.

- Otherwise, computes

$$\mathbf{T}_M = \sum_{i \in I_L} \mathbf{T}_{i,M}$$

with

$$||\mathbf{T}_M|| \leq |I_L| \cdot \sigma\sqrt{m} \cdot H_0(\mathsf{pkag}_{\mathcal{PK}}, M) + |I_{\mathcal{PK}}| \cdot \max_{i \in I_L} ||H_2(\mathsf{pkag}_{\mathcal{PK}}, i)|| + |I_L| \cdot |I_{\mathcal{PK}}| \cdot \sigma^3 m\sqrt{n}.$$

- Generates aggregated subgroup public key

$$\mathsf{spkag}_L = \sum_{i \in I_L} \mathsf{pk}_i.$$

- Returns

$$\mathsf{accmsig}_{\mathcal{PK},L,M} = (\mathbf{T}_M, \mathsf{spkag}_L, \mathsf{pkag}_{\mathcal{PK}}, I_{\mathcal{PK}}, I_L, M).$$

$\mathbf{ASM.vrf}(\mathcal{Y}, \mathsf{accmsig}_{\mathcal{PK},L,M}) \rightarrow (0 \text{ or } 1)$.

- $\mathbf{A} \cdot \mathbf{T}_M = \mathsf{spkag}_L \cdot H_0(\mathsf{pkag}_{\mathcal{PK}}, M) + |I_{\mathcal{PK}}| \cdot \sum_{i \in I_L} \mathbf{A} \cdot H_2(\mathsf{pkag}_{\mathcal{PK}}, i)$ $+\mathsf{pkag}_{\mathcal{PK}} \cdot \sum_{i \in I_L} H_3(i)$,
- $||\mathbf{T}_M|| \leq |I_L| \cdot \sigma\sqrt{m} \cdot H_0(\mathsf{pkag}_{\mathcal{PK}}, M) + |I_{\mathcal{PK}}| \cdot \max_{i \in I_L} ||H_2(\mathsf{pkag}_{\mathcal{PK}}, i)||$ $+|I_L| \cdot |I_{\mathcal{PK}}| \cdot \sigma^3 m\sqrt{n}$.

Theorem 2

The scheme ASM *is unforgeable in the random oracle model if* SIS *problem is hard.*

Thank You!