Simple Schnorr Multi-signatures

Paria Ighanian
Damian Tabaczynski

Supervisor: Rajeev Sahu

Outline

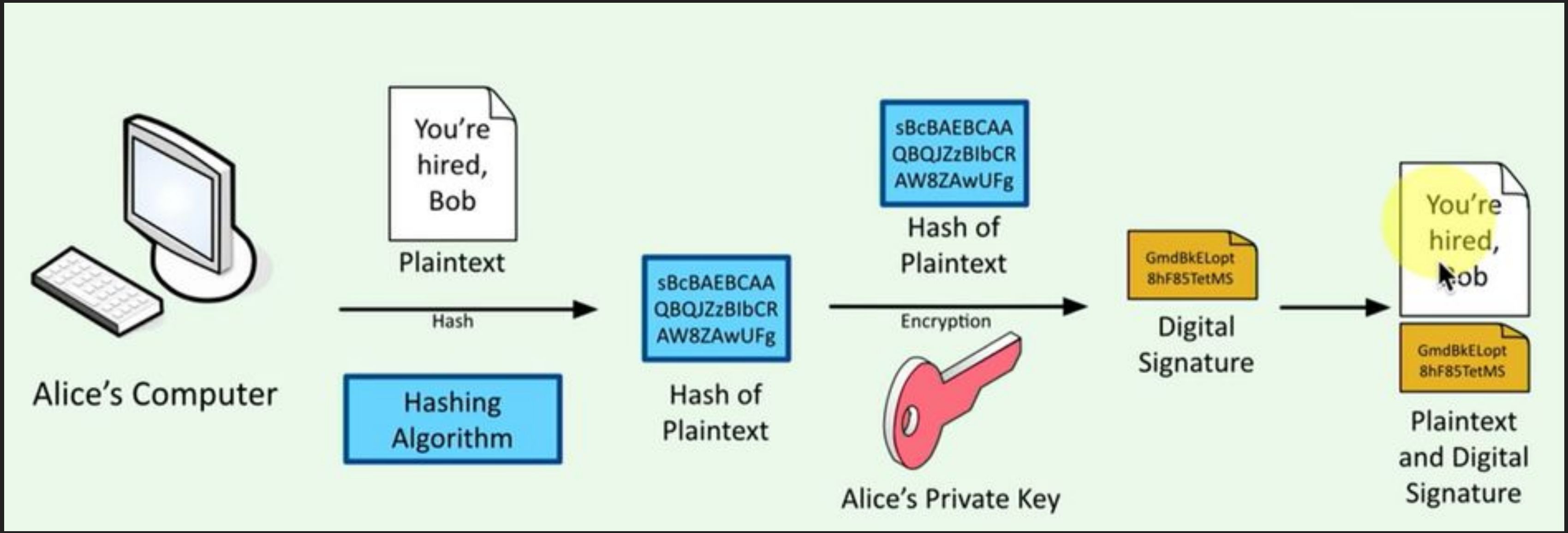
- Digital Signature
- Signature Scheme
- Schnorr Signature
- Multi-signature
- Schnorr Multi-signature
- BN Scheme
- Maxwell et al. Scheme
- Applications to Bitcoin
- Implementation

Digital Signature

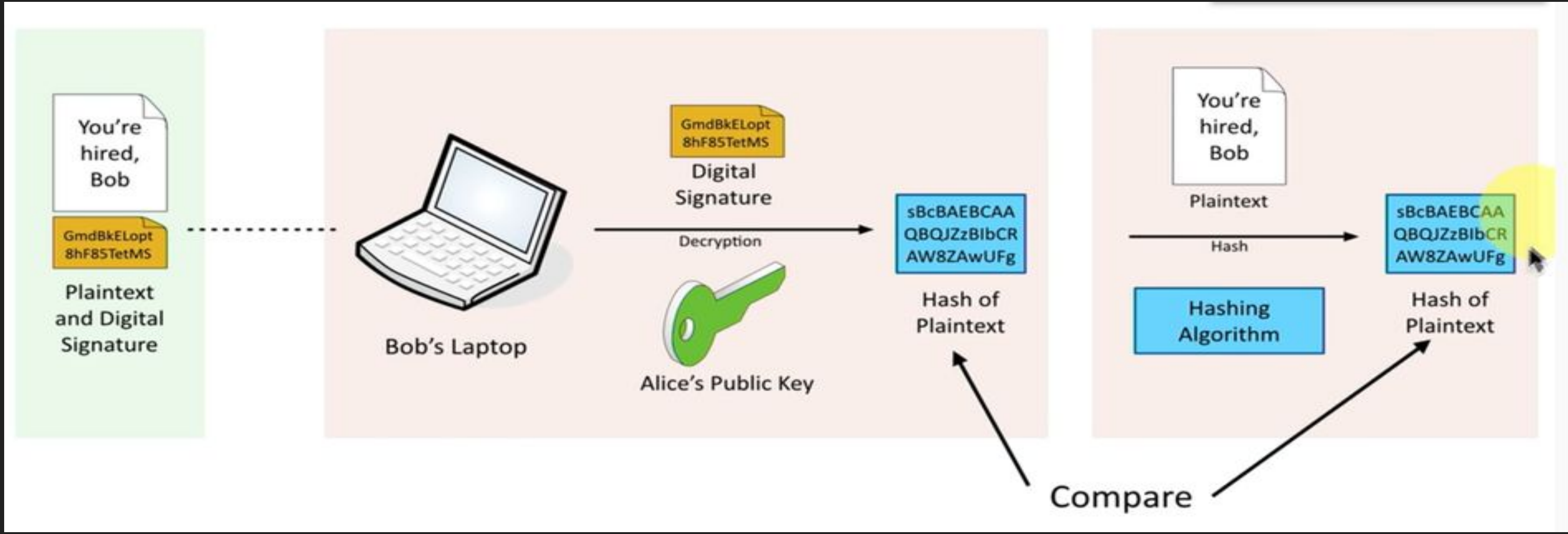
Digital signature is a technique to bind a person to the digital data.

- Authentication. the message was indeed created and sent by the claimed sender (signer)
- Non-repudiation. no signer (party) can deny that it did not sent a message
- Integrity. the message was not altered in transit

Digital Signature. generation



Digital Signature. verification



Signature Scheme

A signature scheme consists of four algorithms:

1. **Setup.** $\mathrm{Setup}(1^\lambda) \to \mathrm{pp}$
2. **Key generation.** $\mathrm{KeyGen}(\mathrm{pp}) \to (\mathrm{pk}, \mathrm{sk})$
3. **Signature.** $\mathrm{Sign}(\mathrm{sk}, m, \mathrm{pp}) \to \sigma$
4. **Verification.** $\mathrm{Ver}(\mathrm{pk}, m, \sigma, \mathrm{pp}) \to \{0,1\}$

Π = (Setup, KeyGen, Sign, Ver) is the signature scheme

Schnorr Signature

A digital signature for which the security is based on the Discrete Logarithm assumption

Setup. A cyclic group $\mathbb{G}$ of prime order p, a generator g of $\mathbb{G}$, a hash function H

Key generation. A private/public key pair (x, X) where $X = g^x$, $x \in \{0, \ldots, p - 1\}$

Signature. To sign a message m, The signer chooses a random integer $r \in \mathbb{Z}_p$, Computes:

- $\mathrm{R} = g^r$,
- $\mathrm{c} = H(X, \mathrm{R}, m)$,
- $\mathrm{s} = r + cx$

The signature is the pair $\sigma = (\mathrm{R}, \mathrm{s})$

Verification. $g^{\mathrm{s}} = \mathrm{R}X^{\mathrm{c}}$

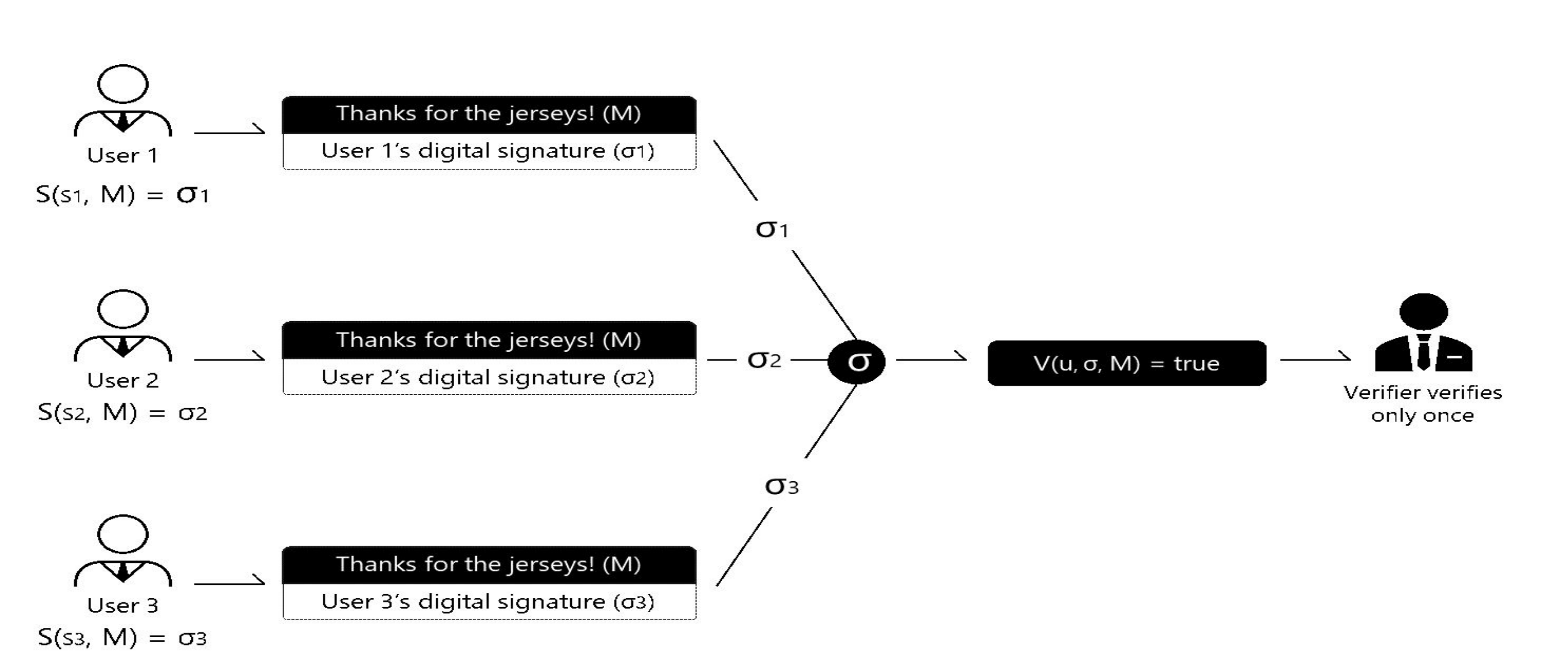
Multi-signature

- A multisignature scheme allows a group of signers to produce a joint signature σ on a common message m
- Concatenating individual signatures is not helpful
- It should be independent from the number of signers

Assume n users with public-private key pairs (pk_i, sk_i)

- Each user i signs m to get a signature σ_i
- All n signatures are combined into one signature σ
- Verification of σ can be done given the message m, and the set of public keys of all signers $pk_1,\ldots,pk_n$

Multi-Signature



Schnorr Multi-signature

A group of n signers want to cosign a message m

Setup. A cyclic group $\mathbb{G}$ of prime order p, a generator g of $\mathbb{G}$, a hash function H,

Key generation. $L = \{ X_1 = g^{x_1}, \ldots, X_n = g^{x_n} \}$

$$X = \Pi_{i=1}^{n} X_i$$

Signature. Each signer generates and shares $R_i = g^{r_i}$

Each signer computes $R = \Pi_{i=1}^{n} R_i$, $c = H(X, R, m)$, $s_i = r_i + cx_i$

$\sigma = (R, s)$ where $s = \Sigma_{i=1}^{n} s_i \bmod p$

Verification. $g^s = RX^c$

Rogue Key Attack

Multisignature schemes have to be secure against the Rogue key attack!

The adversary chooses his public key as a function of public keys of honest users, allowing him to produce forgeries easily.

Let $X_1 = g^{x_1}(\Pi_{i=2}^{n} X_i)^{-1}$ for public keys $\{X_1, \dots, X_n\}$

Two Signers

Alice

Bob

$$L = \{ X_a = g^{x_a}, X_b = g^{x_b} \}$$

$$R = g^{r_a} . g^{r_b} = g^{(r_a + r_b)} = g^{r}$$

$$s_a = r_a + cx_a \ , \ s_b = r_b + cx_b$$

$$c = H(X, R, m), X = g^{x_a} . g^{x_b} = g^{x}$$

$$s = s_a + s_b = (r_a + r_b) + c(x_a + x_b)$$

Two Signers

Alice

Bob

$L = \{ X_b , X'_a = X_a - X_b \}$

$R_b , \; R'_a = R_a - R_b$

$s_b = r_b + c x_b \; , s_a = r_a + c x_a$

$c = H(X, R, m)$

$$\begin{aligned} s_{agg} G &= R_b + R'_a + c\,(X_b + X'_a) \\ &= R_b + (R_a - R_b) + c\,(X_b + X_a - X_b) \\ &= R_a + c\,X_a \\ &= r_a G + c(x_a G) = s_a \end{aligned}$$

$S_{agg} = s_a$

NOT SECURE!

KOSK

Knowledge Of Secret Key:

Proving knowledge of a secret key during public key registration with a certificate authority (CA) in order to prevent Rogue key attack.

Drawback:

- This assumption is not realized by existing public key infrastructures (PKI) i.e. registration protocols specified by the most widely used standards, such as RSA, do not specify that CA's should require proofs of knowledge.

Solution: distinct challenge c_i

The idea is based on Bellare and Neven scheme which is secure in the plain public-key model.

Consider distinct c_i per cosigner $c_i = H(\langle L \rangle, X_i, R, m)$, $s_i = r_i + c_i x_i$

$R = \Pi_{i=1}^{n} R_i$, $\langle L \rangle$ some unique encoding of the multiset of public keys L

Each signer first sends $t_i = H'(R_i)$ to other cosigners

so nobody allows to set $R = \Pi_{i=1}^{r} R_i$

$g^s = R\Pi_{i=1}^{n} X_i^{c_i}$

Bellare and Neven Scheme

Setup. A cyclic group $\mathbb{G}$ of prime order p, a generator g of $\mathbb{G}$, two hash functions H, H'

Key generation. A private/public key pair (x, X) where $X = g^x$, $x \in \{0, \ldots, p - 1\}$

Signature. $\sigma = (R, s)$ where $s_i = r_i + c_i x_i$, $c_i = H(\langle L \rangle, X_i, R, m)$

$R = \Pi_{i=1}^{n} R_i$, $\langle L \rangle$ some unique encoding of the multiset of public keys L

Each signer first sends $t_i = H'(R_i)$ to other cosigners

Verification. $g^s = R\Pi_{i=1}^{n} X_i^{c_i}$

Maxwell Scheme

A variant of the BN scheme with key aggregation in the plain public-key model

$c_i = H_{agg}(\langle L \rangle, X_i).\ H_{sig}(X, R, m) \qquad X = \Pi_{i=1}^{n} X_i^{a_i} \quad a_i = H_{agg}(\langle L \rangle, X_i)$

$g^s = R\Pi_{i=1}^{n} X_i^{a_i c} = RX^c \qquad c = H_{sig}(X, R, m)$

Maxwell Scheme

Setup. A cyclic group $\mathbb{G}$ of prime order p, a generator g of $\mathbb{G}$, three hash functions H_{com}, H_{agg}, H_{sig}

Key generation. A private/public key pair (x, X) where $X = g^x$, $x \in \{0, \ldots, p - 1\}$

Signature. $\sigma = (R, s)$ where $s_i = r_i + c a_i x_i$, $c = H_{\text{agg}}(\langle L \rangle, X_i)$. $H_{\text{sig}}(X, R, m)$

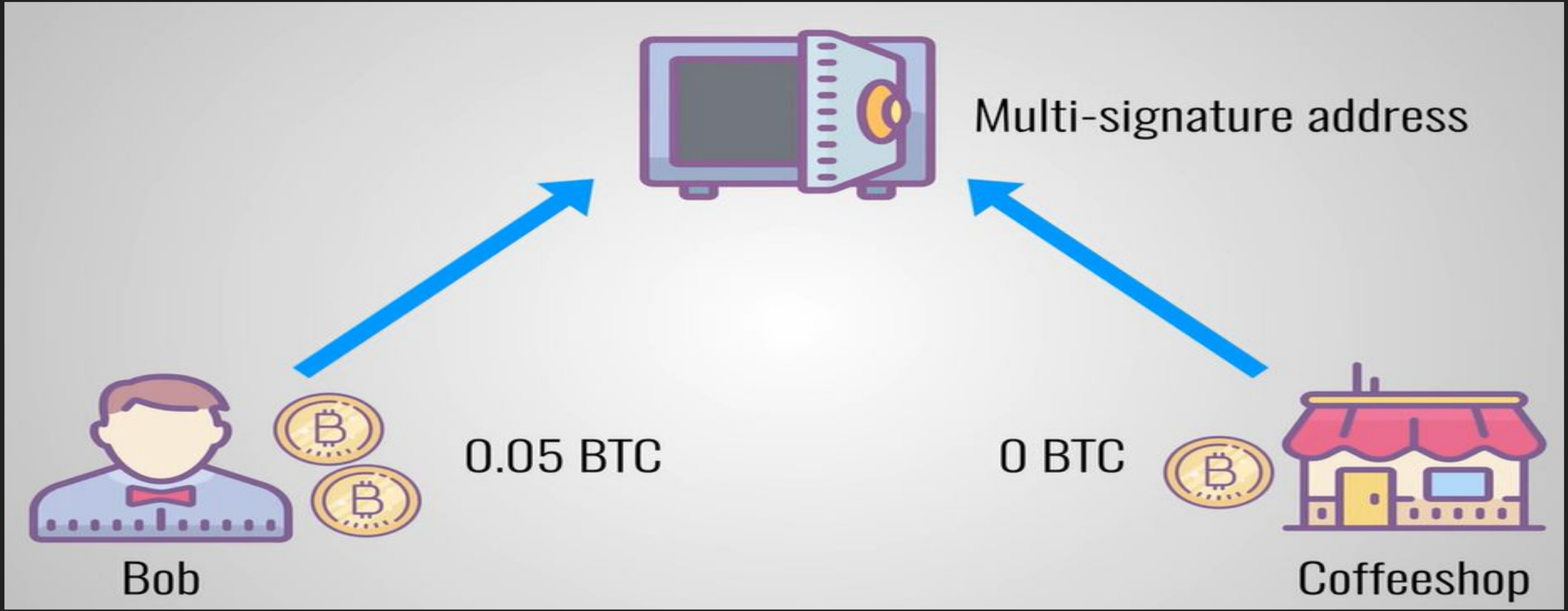
$a_i = H_{\text{agg}}(\langle L \rangle, X_i)$, $X = \Pi_{i=1} X_i^{a_i}$, $R = \Pi_{i=1}^{n} R_i$,

Each signer first sends $t_i = H_{\text{com}}(R_i)$ to other cosigners

Verification. $g^s = R\Pi_{i=1}^{n} X_i^{a_i c}$

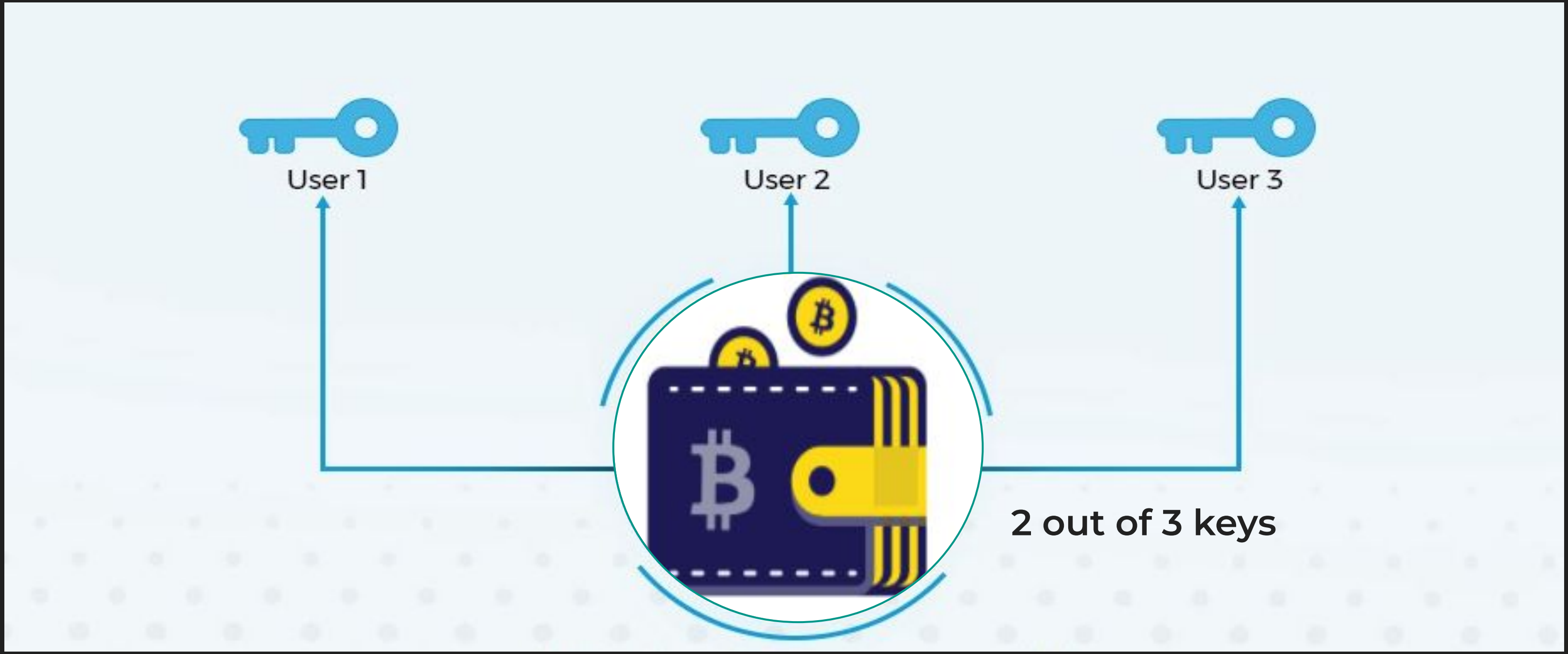
Applications to Bitcoin

- Lightning Network Channel



Applications to Bitcoin

- Multisig wallet



Implementation

Algorithm 1 Setup

q - random prime, k - random number
$p = kq + 1$ that p is prime number
generator $g = h^k \pmod p$, h is random number and $h^k \neq 1 \pmod p$
Output: q, p, g

Algorithm 2 Key Generation

Input: p, q
private key - $x \in (0, p)$
public key - $X = g^x \pmod p$
Output: (x, X)

Algorithm 3 Signature round 1

Input: $L = (X_0, \ldots, X_n)$
for $signer = 1, 2, \ldots, N$ **do**
    $a_i = H_{agg}(L, X_i)$
    $\tilde{X} = \prod_{i=1}^{n} X_i^{ai}$
end for
Output: $\tilde{X}$

Implementation

Algorithm 4 Signature round 2

Input: p, q
for $signer = 1, 2, \ldots, N$ **do**
    $r_i \in (0, q)$
    $R_i = g^{r_i} \pmod{p}$
    Send to every signer $t_i = H_{com}(R_i)$
end for
for $signer = 1, 2, \ldots, N$ **do**
    Send to every signer R_i
end for
for $signer = 1, 2, \ldots, N$ **do**
    If $t_i = H_{com}(R_i)$ true then continue, else abort.
end for

Implementation

Algorithm 5 Signature round 3

Input: $L = (X_0, \ldots, X_n)$, $\tilde{X}, r_i, a_i, x_i, p, q, m$
for $signer = 1, 2, \ldots, N$ **do**
    $R = \prod_{i=1}^{n} R_i$
    $c = H_{sig}(\tilde{X}, R, m)$
    $s_i = r_i + c a_i x_i \pmod q$
end for
$s = \sum_{i=1}^{n} s_i \pmod q$
Output: $\sigma = (R, s)$

Algorithm 6 Verification

Input: $L = (X_0, \ldots, X_n)$, R, s, m, p
Verifier performs:
for $signer = 1, 2, \ldots, N$ **do**
    $a_i = H_{agg}(L, X_i)$
end for
$\tilde{X} = \prod_{i=1}^{n} X_i^{ai}$
$c = H_{sig}(\tilde{X}, R, m)$
Output: if $g^s = R\tilde{X}^c \pmod p$ then true else false

Results

Different Runtime based on the number of signers

Number of signers	Setup time [ms]	Key Generation time [ms]	Signature time [ms]	Verification time [ms]	Overall time [ms]
10	55,99	1,55	16,67	2,44	76,65
30	41,96	3,49	139,65	4,97	190,07
100	51,43	11,27	1950,04	18,18	2030,91
300	70,96	30,71	26757,31	93,43	26952,41

Acknowledgment

We would like to thank our supervisor, Dr. Rajeev Anand SAHU,

for his valuable discussions and feedback.

THANK YOU!