Multi-signer Strong Designated Multi-verifier Signature Schemes based on Multiple Cryptographic Algorithms

Neha Arora, R. K. Sharma

Department of Mathematics, Indian Institute of Technology, New Delhi, 110016, India

Abstract

A designated verifier signature scheme allows a signer to generate a signature that only the designated verifier can verify. This paper proposes multi-signer strong designated multi-verifier signature schemes based on multiple cryptographic algorithms and has proven their security in the random oracle model.

Keywords: Multi-signer, designated multi-verifier, bilinear pairing, factorization, discrete logarithm, Blockchain
2010 Math. Sub. Classification: 94P60, 94A62, 68P25, 68P30

1 Introduction

Digital signature \[5, 6, 16\] is a mathematical algorithm used to validate the authenticity, integrity, and non-repudiation of a message or document. In digital signature algorithms, anyone having the signature and signer’s public key can validate the signature, which is not desired in some cases. Therefore, the undeniable signature \[2\] was introduced by David Chaum and Hans van Antwerpen in 1989. An undeniable signature is a digital signature that requires the signer’s cooperation to verify signatures. However, the validity of an undeniable signature can be ascertained by anyone issuing a challenge to the signer and testing the signer’s response. To solve this problem, Jakobsson Markus, Kazue Sako, and Russell Impagliazzo introduced a designated verifier signature scheme \[8\] in 1996. It provides authentication of a message, and the signer chooses a designated verifier in advance, which makes it non-interactive. However, it does not provide non-repudiation property. Also, in this scheme, the verifier can generate a transcript and convince the third party that the signer created the signature. In the same year, a stronger version is introduced as a strong designated verifier signature scheme. However, in 2003, Guilin Wang designed an attack in \[22\] on the Jakobsson scheme. Later, in 2003, an efficient strong designated verifier signature scheme \[17\] was introduced by Shahrokh Saeednia, Steve Kremer, and Olivier Markowitch. However, in 2007, Ji-Seon Lee and Jik Hyun Chang in \[12\] found that Saeednia scheme \[17\] was not secure, and it would reveal the signer’s identity if the secret key of the signer is compromised. Later, they provide an improved version \[13\]. However, in 2012, Liao Da-jian and Tang Yuan-sheng found that \[13\] does not protect the identity of the signer, and it can be revealed if compromised. Thus, they designed an improved version \[3\] of \[13\] which protects the signer’s identity but lost the security

1 emails: nehaarora1907@gmail.com (Neha), rksharmaiitd@gmail.com (Rajendra)
properties of the designated verifier signature scheme. Also, in 2017, \cite{9} proved that the scheme \cite{13} is not secure, and signature can be forged without knowing the signer’s private key and designed a new strong designated verifier signature scheme.

In 2015, Pankaj Sarde and Amitabh Banerjee reviewed \cite{17} and proposed strong designated verifier signature scheme based on discrete logarithm problem \cite{18}.

In 2018 and later in 2021, Nadiah Shapuan and Eddie Shahril Ismail proposed a strong designated verifier signature scheme with two hard problems: Factoring problem and discrete logarithm problem \cite{19, 20}.

Simultaneously, Few multi signer designated verifier signature scheme \cite{7, 14, 24}, designated multi verifier signature schemes \cite{10, 11, 15, 23} and multi-signer designated multi-verifier signature schemes \cite{1, 4, 21} were introduced.

In the case of Multi-signer designated multi-verifier signature schemes, signatures can be generated in two different ways:

1. All signers come together and cooperate to generate a new signature.

2. All signers generate their signature and make it public to the system, and then, the system generates a new signature using all available signatures.

Multi-signer designated multi-verifier signature schemes has application in various areas. A few of them are mentioned below:

1. Let A be a group of board of Directors of a company, and they have signed some crucial and confidential data which may lead to the increase in profits and price of shares of a company and make its encoded copy public such that each shareholder can verify if the decisions will benefit them or not.

2. **Application using Blockchain Technology:**

To apply our scheme to the blockchain, we made a few assumptions:

(a) Our example best suits a small blockchain where participants are limited, and the transaction information is shared among them only.

(b) Our platform is a smart contract-enabled permissioned Blockchain Network.

(c) Each signer and verifier signed a smart contract, and any cheating and violation led to the blocked account.

(d) Only designated verifiers are allowed to verify and validate the information.

(e) For every transaction information shared, the participants will be the signers, and the remaining will act as verifiers.

(f) Moreover, ownership is not assigned to a single person. Thus, overriding, editing, or deleting the transaction or any other related information is not possible without the consent of a fixed number of participants.

(g) A new block is generated after a fixed predetermined time interval.

(h) The hash value of all transactions in a fixed interval are merged as a root hash in Merkle tree.

(i) Each block is linked with the previous block.
(j) Each block contains block version, Merkle tree root hash, timestamp, nBits, nounce, parent block hash.

| Block Header | Merkle tree root hash | Parent block hash |
|--------------|-----------------------|------------------|
| Block version |                       |                  |
| Timestamp    | nBits                 | Nounce           |

Transaction Counter

TX  TX  TX

Block Structure

We shall show it with an example:

(a) We assume we have 20 participants in our Blockchain Network and each member is given an identity $P_i$, $1 \leq i \leq 20$.

(b) A new block is generated after every $t'$ minutes.

(c) Let $T_{i,j}$ be the transaction id where $i$ represents the block number and $j$ represents the transaction number of $i$-th block.

(d) Let $P_1$ transfers some amount to $P_2, P_3, P_5, P_7$ and $P_9$ transfer some amount to $P_1, P_3, P_5, P_11$ in a fixed time interval $[0, t']$. Let $S_1 = \{P_1, P_2, P_3, P_5, P_7, P_9, B_{11}\}$ be the set of signers. None of them wants to reveal the exact amount, but their ids, and encoded value of the change in their respective balance. In this case, each signer creates and sends the signature to the system. The system generates a new signature and broadcasts it to all remaining participants, and each verifier, within a specific time, can verify and validate the transaction.

(e) Let $P_7$ transfer some amount to $P_{13}, P_{15}$, $P_4$ to $P_2, P_7, P_{13}$, and $P_3$ to $P_4, P_5, P_{11}, P_{13}$ in a fixed time interval $[t', 2t']$. Then, $S_2 = \{P_2, P_3, P_4, P_5, P_7, P_{11}, P_{13}, P_{15}\}$ be the new set of signers and remaining participants will act as verifiers and they can verify and validate the transactions with a specific time.
This way, we can record and verify multiple transactions compactly. Also, the privacy of each transaction can be maintained by keeping individual amounts secret and making hash value the total amount public. It also decreases the size of the blocks and reduces the storage cost, making it cheaper to operate.

So far, we have seen a few multi-signer designated verifier signature schemes, designated multi-verifier signature schemes, and multi-signer designated multi-verifier signature schemes based on either single hard problem or bilinear pairing. This paper proposes two individual multi-signer strong designated multi-verifier signature schemes and combines them to enhance their security. Our final scheme is based on multiple hard problems, and it requires the authorization of each participating signer. Also, our scheme is non-interactive as the system will act as an intermediator. Later, we showed that our scheme is a strong, unforgeable, non-transferable designated verifier scheme.

This paper is organized as follows: In Section 2, we define factoring problem, discrete logarithm problem, bilinear pairing, strong designated verifier, strong designated verifier signature scheme, and multi-signer designated multi-verifier signature schemes. Section 3 is divided into four subsections. Section 3.1 is an extension of [24] to multi-verifier scheme, and it is based on bilinear pairing. Section 3.2 is an extension of [19, 20] to multi-signer multi-verifier scheme, and it is based on two hard problems: factoring problem and discrete logarithm problem. Section 3.3 is an improvisation of section 3.2 based on the elliptic curve. Section 3.4 gives a glimpse of the combination of the first and second algorithm. In section 4, we have done the security analysis of our scheme.

2 Preliminaries

Definition 1. Factoring problem : Let $n$ be a large positive composite integer such that $n = pq$, where $p$ and $q$ are large primes. Then, solving to find the factorization of $n$ is called factoring problem.

Definition 2. Discrete Logarithm problem: Let $p, q$ be primes such that $q$ divides $p - 1$. Let $G$ be a multiplicative group of order $p$, and $g \in G$ such that $|g| = q$ and $y \equiv g^x \pmod{n}$, then for given $g$ and $y$, computation of $x$ is called as discrete logarithm problem.

Definition 3. Bilinear pairing :

Let $G_1$ and $G_2$ be two cyclic groups (w.r.t. addition) of prime order $p$ with generators $P_1$ and $P_2$, respectively. Define a map $e : G_1 \times G_2 \rightarrow G_T$, where $G_T$ is a group (w.r.t. multiplication) of order $p$. Then, the map $e$ is s.t.b. $\mathbb{F}_p$-bilinear if $e(aP_1, bP_2) = e(P_1, P_2)^{ab}$ $\forall a, b \in \mathbb{F}_p$, non-degenerate if $e(P_1, P_2) \neq 1_{G_T}$, and symmetric if $G_1 = G_2$. 
The pairing \( e \) is an admissible bilinear map if it is non-degenerate and it can be efficiently computable.

**Definition 4.** \([T] \) **Strong designated Verifier**: Let \( P(S,V) \) be a protocol for signer \( S \) to prove the truth of the statement \( \omega \) to verifier \( V \). We say that \( P(S,V) \) is strong designated verifier proof if anybody can produce identically distributed transcripts that are indistinguishable from those of \( P(S,V) \) for everybody, except for Verifier \( V \).

**Definition 5.** **Strong Designated Verifier Signature Scheme**: A strong designated verifier signature scheme consists of the following algorithms:

1. **Set up**: \( sp \leftarrow \text{Setup}(K) \), where \( K \) is a security parameter, \( sp \) is system parameters.

2. **Key generation**: \((sk_U, pk_U) \leftarrow \text{KeyGen}(sp, K)\), where \( sk_U \) and \( pk_U \) are the secret key and public key of user \( U \) respectively.

3. **Signature Algorithm** \( \text{Sign}_S \rightarrow D \):
   
   \[ \text{sign}_S \leftarrow \text{Sign}_S \rightarrow D(M, sk_S, pk_D) \]

   where \( M \in \{0,1\}^* \) is the message to be signed, \( sk_S \) denote the private key of signer \( S \) and \( pk_D \) denote the public key of designated verifier \( D \).

4. **Verification Algorithm** \( \text{Verify}_D \):
   
   \[ \{\text{accept, reject}\} \leftarrow \text{Verify}_D(M, sk_D, pk_S, \text{sign}_S) \]

   where \( sk_D \) denote the secret key of designated verifier \( D \) and \( pk_S \) denote the public key of signer \( S \).

**Definition 6.** **Multi-signer strong designated Multi-verifier signature scheme**: A Multi-signer strong designated Multi-verifier signature scheme consists of the following algorithms:

1. **Set up**: \( sp \leftarrow \text{Setup}(K) \), where \( K \) is a security parameter, \( sp \) is system parameters.

2. **Key generation**: \((sk_U, pk_U) \leftarrow \text{KeyGen}(sp, K)\), where \( sk_U \) and \( pk_U \) are the secret key and public key of user \( U \) respectively.

3. **Signature Algorithm** \( \text{Sign}(S_1, S_2, \ldots, S_n) \rightarrow (D_1, D_2, \ldots, D_m) \):
   
   \[ \text{sign}_A \leftarrow \text{Sign}(S_1, S_2, \ldots, S_n) \rightarrow (D_1, D_2, \ldots, D_m)(M, sk_{S_1}, sk_{S_2}, \ldots, sk_{S_n}, pk_{D_1}, pk_{D_2}, \ldots, pk_{D_m}) \]

   where \( M \in \{0,1\}^* \) is the message to be signed, \((sk_{S_1}, sk_{S_2}, \ldots, sk_{S_n})\) denote the private key of \( n \) signers \((S_1, S_2, \ldots, S_n)\) and \((pk_{D_1}, pk_{D_2}, \ldots, pk_{D_m})\) denote the public key of \( m \) designated verifiers \((D_1, D_2, \ldots, D_m)\).

4. **Verification Algorithm** \( \text{Verify}(D_1, D_2, \ldots, D_m) \):
   
   \[ \{\text{accept, reject}\} \leftarrow \text{Verify}(D_1, D_2, \ldots, D_m)(M, sk_{D_1}, sk_{D_2}, \ldots, sk_{D_m}, pk_{S_1}, pk_{S_2}, \ldots, pk_{S_n}, \text{sign}_A) \]

   where \((sk_{D_1}, sk_{D_2}, \ldots, sk_{D_m})\) denote the secret key of \( m \) designated verifiers \((D_1, D_2, \ldots, D_m)\) and \((pk_{S_1}, pk_{S_2}, \ldots, pk_{S_n})\) denote the public key of \( n \) signers \((S_1, S_2, \ldots, S_n)\).
3 Proposed Schemes

This paper defines a secure, strong designated signature scheme with multi-signers and multi-verifiers based on multiple hard problems. Our scheme is divided into four sub-sections. Section 3.1 is an extension of [24] to multi-verifier scheme, and it is based on bilinear pairing. Section 3.2 is an extension of [19, 20] to multi-signer multi-verifier scheme, and it is based on two hard problems: factoring problem and discrete logarithm problem. Section 3.3 is an improvisation of section 3.2 based on the elliptic curve. We have illustrated the algorithm with examples. We can combine the first algorithms with the second algorithm or an improvised version of the second algorithm by making minor changes described in section 3.4.

Let $A = \{S_1, S_2, ..., S_n\}$ be the group of signers and $B = \{D_1, D_2, ..., D_m\}$ be the group of designated verifiers. Here $A$ and $B$ are systems that stores information which is made public by $S_i$ and $D_j$ respectively and then they interact with each other $l$ times, where $l$ is a predetermined number.

3.1 Algorithm 1

Set up

Let $K$ be the security parameter and $sp = (G, G_\tau, g, p, e, H_1) \leftarrow \text{Setup}(K)$, where

- $G$ is an additive cyclic group of large prime order $p$ and $g$ is a generator of $G$,
- $G_\tau$ is a multiplicative cyclic group of prime order $p$,
- $e : G \times G \rightarrow G_\tau$ is symmetric admissible bilinear pairing map,
- $H_1 : \{0,1\}^* \rightarrow G$ is a cryptographically secure hash function.

Key Generation

- Let there are $n$ signers $\{S_1, S_2, ..., S_n\}$ and $m$ designated verifiers $\{D_1, D_2, ..., D_m\}$
- Let $M \in \{0,1\}^*$ be the message, then $H_1(M) \in G$.
- Let $a_i$ be the secret key of signer $S_i$ and $p_i = a_i g$ be the corresponding public key of $S_i \ \forall \ i \in \{1,2,\ldots,n\}$ known to system $A$.
- System $A$, then, publishes $u = \sum_{i=1}^{n} p_i$.
- Let $b_i$ be the secret key of designated verifier $D_i$ and $q_i = b_i g$ be the corresponding public key of $D_i \ \forall \ i \in \{1,2,\ldots,m\}$ known to system $B$.
- System $B$, then, publishes $v = \sum_{i=1}^{m} q_i$.

Signature Algorithm
• Each signer $S_i$ computes $\sigma_i = \text{Sign}(M, a_i, q_1, q_2, \ldots, q_m) = e(H_1(M), a_i v)$ as his signature and make it public.

• After all signatures $(\sigma_1, \sigma_2, \ldots, \sigma_n)$ has been collected by the system $A$, it computes $\sigma = \prod_{i=1}^{n} \sigma_i$ and send it to system $B$.

Verification Algorithm

• System $B$ has a message $M$ and $\sigma$.

• Each designated verifier $D_i$ computes $\zeta_i = \text{Sign}(M, b_i, p_1, p_2, \ldots, p_n) = e(H_1(M), b_i u)$ and make it public to the system $D$.

• System $B$ computes $\zeta = \prod_{i=1}^{m} \zeta_i$.

• Accept the output if $\sigma \equiv \zeta$ in $G_\tau$, otherwise reject.

Transcript-Simulation

Since, $\sigma \equiv \zeta$ in $G_\tau$, simulated signature is equivalent to the signature produced by system A, thus simulation in this case is obvious.

Example 1

Let, $G = \mathbb{Z}_{11}$ is an additive cyclic group of prime order 11 and 2 is a generator of $G$, $G_\tau = \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}$ $\bigodot_{23}$ is a multiplicative cyclic group of prime order 11, $\text{e} : G \times G \rightarrow G_\tau$ is symmetric admissible bilinear pairing map such that $\text{e}(2, 2) = 2$.

Then, for any $a, b \in G$, $\exists \ 0 \leq x, y \leq 10$ respectively such that $a = 2x$ and $b = 2y$, implies $\text{e}(a, b) = \text{e}(2, 2)^{xy} = 2^{xy}$

Let, $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_{11}$ is a cryptographically secure hash function and $M \in \{0, 1\}^*$ be the message, then $H_1(M) \in \mathbb{Z}_{11} \Rightarrow H_1(M) = 2c$ for some $1 \leq c \leq 11$.

Let $A = \{S_1, S_2, S_3\}$ be three signers and $B = \{D_1, D_2\}$ be two designated verifiers.

Let $a_1 = 3 \ , a_2 = 7 \ , a_3 = 9$ be the secret keys and $p_1 = 6 \ , p_2 = 3 \ , p_3 = 7$ be public keys (made public to system $A$ only) of $S_1, S_2, S_3$ respectively. Then System $A$ will compute $u = \sum_{i=1}^{3} p_i \equiv 5$ and make it public.

Let $b_1 = 6 \ , b_2 = 8$ be the secret keys and $q_1 = 1 \ , q_2 = 5$ be public keys (made public to system $B$ only) of $D_1, D_2$ respectively. Then System $B$ will compute $v = \sum_{i=1}^{2} q_i \equiv 6$ and make it public.

Note that

$$\sigma_1 = e(2, 2)^{9c} = 2^{9c}$$
$$\sigma_2 = e(2, 2)^{10c} = 2^{10c}$$
$$\sigma_3 = e(2, 2)^{5c} = 2^{5c}$$
implies
\[ \sigma = \prod_{i=1}^{3} \sigma_i = 2^{24c} \equiv 2^{2c} \text{ in } G_{\tau} \]

Similarly
\[ \zeta_1 = e(2, 2)^{4c} = 2^{4c} \]
\[ \zeta_2 = e(2, 2)^{9c} = 2^{9c} \]

implies
\[ \zeta = \prod_{i=1}^{2} \zeta_i = 2^{13c} \equiv 2^{2c} \text{ in } G_{\tau} \]

**Example 2**

Let, \( G = \mathbb{Z}_{53} \) is an additive cyclic group of prime order 53 and \( g = 5 \) is a generator of \( G \), \( G_{\tau} = < 3 / 3^{53} = 1 > \otimes_{107} \) is a multiplicative cyclic group of prime order 53, \( e : G \times G \rightarrow G_{\tau} \) is symmetric admissible bilinear pairing map such that \( e(5, 5) = 3 \).

Then, for any \( a, b \in G \), \( \exists 0 \leq x, y \leq 52 \) respectively such that \( a = 5^x \) and \( b = 5^y \), implies \( e(a, b) = e(5, 5)^{xy} = 3^{xy} \)

Let, \( H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_{53} \) is a cryptographically secure hash function and \( M \in \{0, 1\}^* \) be the message, then \( H_1(M) \in \mathbb{Z}_{53} \Rightarrow H_1(M) = 5c \) for some \( 1 \leq c \leq 52 \).

Let \( A = \{S_1, S_2, S_3, S_4, S_5\} \) be the system having five signers and \( B = \{D_1, D_2, D_3, D_4, D_5, D_6, D_7\} \) be the system having seven designated verifiers.

Let \( a_1 = 7, a_2 = 12, a_3 = 15, a_4 = 19, a_5 = 31 \) be the secret keys and \( p_1 = 35, p_2 = 7, p_3 = 22, p_4 = 42, p_5 = 49 \) be public keys (made public to system \( A \) only) of \( S_1, S_2, S_3, S_4, S_5 \) respectively. Then System \( A \) will compute \( u = \sum_{i=1}^{5} p_i \equiv 49 \) and make it public.

Let \( b_1 = 10, b_2 = 13, b_3 = 17, b_4 = 23, b_5 = 51, b_6 = 27, b_7 = 36 \) be the secret keys and \( q_1 = 50, q_2 = 12, q_3 = 32, q_4 = 9, q_5 = 43, q_6 = 29, q_2 = 21 \) be public keys (made public to system \( B \) only) of \( D_1, D_2 \) respectively. Then System \( B \) will compute \( v = \sum_{i=1}^{6} q_i \equiv 37 \) and make it public.

Note that
\[ \sigma_1 = 3^{20c} \]
\[ \sigma_2 = 3^{4c} \]
\[ \sigma_3 = 3^{5c} \]
\[ \sigma_4 = 3^{24c} \]
\[ \sigma_5 = 3^{28c} \]

implies
\[ \sigma = \prod_{i=1}^{5} \sigma_i \equiv 3^{28c} \text{ in } G_{\tau} \]

Similarly
\[ \zeta_1 = 3^{45c} \]
\[\zeta_2 = 3^{32c} \]
\[\zeta_3 = 3^{50c} \]
\[\zeta_4 = 3^{24c} \]
\[\zeta_5 = 3^{44c} \]
\[\zeta_6 = 3^{42c} \]
\[\zeta_7 = 3^{3c} \]

implies

\[\zeta = \prod_{i=1}^{7} \zeta_i \equiv 3^{28c} \text{ in } G_r\]

### 3.2 Algorithm 2

#### Set up

Let \( K \) be the security parameter and \( sp = (\tilde{G}, g_A, g_B, p, n_A, n_B, H_2) \leftarrow \text{Setup}(K) \), where

- \( p \) be a large prime such that \( n_A \) and \( n_B \) are factors of \( p - 1 \),
- \( \tilde{G} = Z_p^* \) and \( g_A, g_B \in \tilde{G} \) such that \(|g_A| = n_A, |g_B| = n_B|,
- \( H_2 \) be a cryptographically secured hash function with arbitrary bit length.

#### Key Generation

**Key generation algorithm for system \( A \):**

1. System \( A \) chooses prime \( p_A \) and \( q_A \), then computes \( n_A = p_A \ast q_A \).
2. Each \( S_i \) follow the following steps:
   - randomly choose \( e_{A_i} \) such that \( \gcd(e_{A_i}, \phi(n_A)) = 1 \).
   - compute \( d_{A_i} \) such that \( e_{A_i} \cdot d_{A_i} = 1 \pmod{\phi(n_A)} \).
   - choose an integer \( x_{A_i} \in Z_p^* \).
   - calculate \( y_{A_i} = g_A^{x_{A_i}} \pmod{p} \).
   - public key of \( S_i \) is \((e_{A_i}, y_{A_i})\).
   - private key of \( S_i \) is \((d_{A_i}, x_{A_i})\).
3. Note: Since, each member in system \( A \) knows \( \phi(n_A) \) and \( e_{A_i} \) \( \forall i \), they can compute \( d_{A_i} \). Thus, \( d_{A_i} \) is private for members of system \( B \) but not for members of system \( A \).

**Key generation algorithm for system \( B \):**

1. System \( B \) chooses prime \( p_B \) and \( q_B \), then computes \( n_B = p_B \ast q_B \).
2. Each \( D_i \) follow the following steps:
• randomly choose $e_{B_i}$ such that $\text{g.c.d.}(e_{B_i}, \phi(n_{B_i})) = 1$.
• compute $d_{B_i}$ such that $e_{B_i}d_{B_i} = 1 \pmod{\phi(n_{B_i})}$.
• choose an integer $x_{B_i} \in \mathbb{Z}_p^*$.
• calculate $y_{B_i} = g^{x_{B_i}}_{B_i} \pmod{p}$.
• public key of $D_i$ is $(e_{B_i}, y_{B_i})$.
• private key of $D_i$ is $(d_{B_i}, x_{B_i})$.

3. Note : Since, each member in system $B$ knows $\phi(n_{B_i})$ and $e_{B_i}$, $\forall i$, they can compute $d_{B_i}$. Thus, $d_{B_i}$ is private for members of system $A$ but not for members of system $B$.

**Signature Algorithm**

System $A$ generates a signature $(r, s, t, \bar{u})$ for a message $M$ as follows:

1. Each $S_i$ chooses a random integer $k_i$ and keep it secret.
2. Each $S_i$ generates $r_i = g^{k_i}_{A}(y_{B_1}y_{B_2} \ldots y_{B_m})^{-k_i} \pmod{p}$, $s_i = g^{k_i}_{B} \pmod{p}$ and $w_i = g^{k_i}_{A} \pmod{p}$ and make it public to the system $A$.
3. System computes :
   \begin{itemize}
   \item $r = \prod_{i=1}^{n} r_i \pmod{p}$
   \item $s = (\prod_{i=1}^{n} s_i)^r \pmod{p}$
   \item $w = (\prod_{i=1}^{n} w_i)^r \pmod{p}$
   \item $z = H_2(M, w)$
   \item $t = \prod_{i=1}^{m} e_{B_i} \pmod{n_{B_i}}$
   \end{itemize}
4. After computing $(r, s, w, z, t)$, they are made public to all $S_i$ and then each $S_i$ computes $v_i = zx_{A_i} + k_ir$ and make it public to the system $A$.
5. System computes
   \begin{itemize}
   \item $\bar{v} = \sum_{i=1}^{n} v_i \pmod{n_A}$
   \item $\bar{u} = (\bar{v})^{i=1} \pmod{n_A}$
   \end{itemize}

System $A$ sends the message $M$ and the signature $(r, s, t, \bar{u})$ to the system $B$ (and all the designated verifiers $D_i$).
Verification Algorithm

Upon receiving the message \( M \) and signature \((r, s, t, \bar{u})\), each designated verifier \( D_i \) computes \( z_i = s^{x_{Bi}} \) and make it public. Then, each \( D_i \) can separately or together verify and validate the signature by checking the following equations:

1. Computes \( a = (\bar{u})_{i=1}^{n} (mod \ n_A) \)

2. Computes \( b = \prod_{i=1}^{m} d_{Bi} (mod \ n_B) \)

3. Check \( c = g_A^a (y_{A1}y_{A2}...y_{An})^{-b} (mod \ p) = r^r \prod_{i=1}^{m} z_i (mod \ p) \)

4. Lastly, signatures are accepted if \( b = H_2(M, c) \).

5. If any of the condition from above two conditions fail, verifier can deny to accept the signature. In case of more than 2 verifiers, if 50\% or more verifiers deny the signature, system \( B \) will return the signature to system \( A \).

Transcript-Simulation

System \( B \) (or designated verifier \( D \)) can simulate the correct transcripts as follows:

1. System \( B \) chooses a random integer \( k' \) and keep it secret.

2. Then \( D \) generates:
   - \( r' = g_B^{k'} (y_{A1}y_{A2}...y_{An})^{-k'} (mod \ p) \)
   - \( s' = g_A^{k'r'} (mod \ p) \)
   - \( w' = g_B^{k'r'} (mod \ p) \)
   - \( z' = H_2(M, w') \)
   - \( t' = z'_{i=1}^{n} (mod \ n_A) \)
   - \( v' = \sum_{i=1}^{m} v_i' (mod \ n_B) \), where \( v_i' = z' x_{Bi} + k'r' \) is made public by \( D_i \)
   - \( u' = (v')^{d_B} (mod \ n_B) \), where \( d_B = \prod_{i=1}^{m} d_{Bi} \)

System \( B \) simulated signature is \((r', s', t', u')\).
Example 1

Let $A = \{S_1, S_2, S_3\}$ be the group of three signers (or provers) and $B = \{D_1, D_2\}$ be the group of two designated verifiers.

Let $p = 211$ be a prime such that $n_A = 15$ and $n_B = 14$ are factors of $p - 1 = 210$.

Let $\tilde{G} = \mathbb{Z}_{211}$, $g_A = 137$, $g_B = 63$ in $\tilde{G}$ such that $|137| = 15$, $|63| = 14$.

Let $H_2$ be a cryptographically secured hash function with arbitrary bit length.

Key generation algorithm for system $A$:

1. System $A$ chooses prime $p_A = 3$ and $q_A = 5$, then computes $n_A = 3 \times 5$.
2. Each $S_i$ follow the following steps:

|   | $e_{A_i}$ | $d_{A_i}$ | $x_{A_i}$ | $y_{A_i}$ |
|---|-----------|-----------|-----------|-----------|
| $S_1$ | 13        | 5         | 7         | 150       |
| $S_2$ | 7         | 7         | 16        | 137       |
| $S_3$ | 11        | 3         | 21        | 55        |

3. public key of $S_i$ is $(e_{A_i}, y_{A_i})$.
4. private key of $S_i$ is $(d_{A_i}, x_{A_i})$.
5. Note: Since, each member in system $A$ knows $\phi(n_A)$ and $e_{A_i}$, $\forall i$, they can compute $d_{A_i}$. Thus, $d_{A_i}$ is private for members of system $B$ but not for members of system $A$.

Key generation algorithm for system $B$:

1. System $B$ chooses prime $p_B = 2$ and $q_B = 7$, then computes $n_B = 2 \times 7$.
2. Each $D_i$ follow the following steps:

|   | $e_{B_i}$ | $d_{B_i}$ | $x_{B_i}$ | $y_{B_i}$ |
|---|-----------|-----------|-----------|-----------|
| $D_1$ | 5         | 5         | 19        | 153       |
| $D_2$ | 11        | 5         | 17        | 12        |

3. public key of $D_i$ is $(e_{B_i}, y_{B_i})$.
4. private key of $D_i$ is $(d_{B_i}, x_{B_i})$.
5. Note: Since, each member in system $B$ knows $\phi(n_B)$ and $e_{B_i}$, $\forall i$, they can compute $d_{B_i}$. Thus, $d_{B_i}$ is private for members of system $A$ but not for members of system $B$.

Signature Algorithm

1. Each $S_i$ chooses a random integer $k_i$ and keep it secret and then, generates

$$r_i = g_{A_i}^{k_i}(y_{B_1} y_{B_2} \cdots y_{B_e})^{-k_i} \pmod{p},$$

$$s_i = g_{B_i}^{k_i} \pmod{p}$$

and $w_i = g_{A_i}^{k_i} \pmod{p}$

and make it public to the system $A$. 

12
2. System computes:

- \( r = \prod_{i=1}^{3} r_i \equiv 30 \pmod{p} \)
- \( s = (\prod_{i=1}^{3} s_i)^r \equiv 144 \pmod{p} \)
- \( w = (\prod_{i=1}^{3} w_i)^r \equiv 1 \pmod{p} \)
- \( z = H_2(M, w) \)
- \( t = z^{i=1} \equiv z^{55} \equiv z \pmod{14} \)

3. After computing \((r, s, w, z, t)\), they are made public to all \(P_i\) and then each \(P_i\) computes \(v_i = z x_A_i + k_i r\) and make it public to the system \(A\).

\[ v_1 = 7z + 240, \quad v_2 = 16z + 360, \quad v_3 = 21z + 420 \]

4. System computes

- \( \bar{v} = \sum_{i=1}^{3} v_i \equiv 44z \pmod{15} \)
- \( \bar{u} = (\bar{v})^{i=1} \equiv (\bar{v})^{5 \times 7 \times 3} \equiv \bar{v} \pmod{15} \)

System \(A\) sends the message \(M\) and the signature \((r, s, t, \bar{u})\) to the system \(B\) (and all the verifiers \(D_i\)).

**Verification Algorithm**

Upon receiving the message \(M\) and the signature \((r, s, t, \bar{u})\), each designated verifier \(D_i\) computes \(z_i = s^{x_B_i}\) and make it public. Then, each \(D_i\) can separately or together verify and validate the signature by checking the following equations:

\[ \prod_{i=1}^{3} e_{A_i}^{e_{A_i}} \]

1. Computes \( a = (\bar{u})^{i=1} \pmod{n_A} \)
2. Computes \( b = \prod_{i=1}^{2} d_{B_i} \pmod{n_B} \)

3. Check \( c = g_A^a (y_A_1 y_A_2 y_A_3)^b \pmod{p} = r^r \prod_{i=1}^{2} z_i \pmod{p} \)

4. Lastly, signatures are accepted if \( b = H_2(M, c) \).

5. If any of the condition from above two conditions fail, verifier can deny to accept the signature. In case of more than 2 verifiers, if 50% or more verifiers deny the signature, system \( B \) will return the signature to system \( A \).

**Example 2**

Let \( A = \{S_1, S_2, S_3, S_4, S_5\} \) be the group of five signers (or provers) and \( B = \{D_1, D_2, D_3, D_4, D_5, D_6, D_7\} \) be the group of seven designated verifiers.

Let \( p = 102103 \) be a prime such that \( n_A = 91 \) and \( n_B = 187 \) are factors of \( p - 1 = 102102 \). Let \( G = \mathbb{Z}_{102103}^* \) and \( g_A = 44494, g_B = 12733 \) in \( G \) such that \( |44494| = 91, |12733| = 187 \). Let \( H_2 \) be a cryptographically secured hash function with arbitrary bit length.

**Key generation algorithm for system \( A \):**

1. System \( A \) chooses prime \( p_A = 7 \) and \( q_A = 13 \), then computes \( n_A = 7 \times 13 \).

2. Each \( S_i \) follow the following steps:

| \( S_i \) | \( e_{A_i} \) | \( d_{A_i} \) | \( x_{A_i} \) | \( y_{A_i} \) |
|---|---|---|---|---|
| \( S_1 \) | 5 | 29 | 15 | 58327 |
| \( S_2 \) | 11 | 59 | 19 | 69494 |
| \( S_3 \) | 7 | 31 | 24 | 69058 |
| \( S_4 \) | 23 | 47 | 18 | 88515 |
| \( S_5 \) | 35 | 35 | 32 | 26123 |

3. public key of \( S_i \) is \((e_{A_i}, y_{A_i})\).

4. private key of \( S_i \) is \((d_{A_i}, x_{A_i})\).

5. Note : Since, each member in system \( A \) knows \( \phi(n_A) \) and \( e_{A_i} \ \forall i \), they can compute \( d_{A_i} \). Thus, \( d_{A_i} \) is private for members of system \( B \) but not for members of system \( A \).

**Key generation algorithm for system \( A \):**

1. System \( B \) chooses prime \( p_B = 11 \) and \( q_B = 17 \), then computes \( n_B = 11 \times 17 \).

2. Each \( D_i \) follow the following steps:
3. public key of $D_i$ is $(e_{B_i}, y_{B_i})$.

4. private key of $D_i$ is $(d_{B_i}, x_{B_i})$.

5. Note: Since, each member in system $B$ knows $\phi(n_B)$ and $e_{B_i} \forall i$, they can compute $d_{B_i}$. Thus, $d_{B_i}$ is private for members of system $A$ but not for members of system $B$.

### Signature Algorithm

1. Each $S_i$ chooses a random integer $k_i$ and keep it secret and then, generates

$$
    r_i = g_A^{k_i}(y_{B_1} y_{B_2} \ldots y_{B_e})^{-k_i} \pmod{p},
    s_i = g_B^{k_i} \pmod{p} \text{ and } w_i = g_A^{k_i} \pmod{p}
$$

and make it public to the system $A$.

| $S_i$ | $k_i$  | $r_i$  | $s_i$  | $w_i$  |
|------|-------|-------|-------|-------|
| $S_1$| 42980 | 22513 | 68227 | 59022 |
| $S_2$| 68841 | 77234 | 67990 | 84473 |
| $S_3$| 82718 | 60319 | 8171  | 15368 |
| $S_4$| 90739 | 49375 | 58517 | 68284 |
| $S_5$| 19344 | 40471 | 35552 | 91619 |

2. System computes:

- $r = \prod_{i=1}^{5} r_i \equiv 41707 \pmod{p}$
- $s = \left(\prod_{i=1}^{5} s_i\right)^r \equiv 90653 \pmod{p}$
- $w = \left(\prod_{i=1}^{5} w_i\right)^r \equiv 91371 \pmod{p}$
- $z = H_2(M,w)$
- $t = \prod_{i=1}^{7} e_{B_i} \equiv z^{549972423} \equiv z^{103} \pmod{187}$, provided $\text{g.c.d.}(z, 187) = 1$
3. After computing \((r, s, w, z, t)\), they are made public to all \(S_i\) and then each \(S_i\) computes \(v_i = zx_{A_i} + k_ir\) and make it public to the system \(A\).

\[v_1 = 15z+42980r, \quad v_2 = 19z+68841r, \quad v_3 = 24z+82718r, \quad v_4 = 18z+90739r, \quad v_5 = 32z+19344r\]

4. System computes

\[\bar{v} = \sum_{i=1}^{5} v_i \equiv 17z + 31 \pmod{91}\]

\[\bar{u} = (\bar{v})^{\prod_{i=1}^{5} d_{A_i}} \equiv (\bar{v})^{87252445} \equiv (\bar{v})^{37} \pmod{91}\]

System \(A\) sends the message \(M\) and the signature \((r, s, t, \bar{u})\) to the system \(B\) (and all the verifiers \(D_i\)).

**Verification Algorithm**

Upon receiving the message \(M\) and the signature \((r, s, t, \bar{u})\), each verifier \(D_i\) computes \(z_i = s^{x_{B_i}}\) and make it public. Then, each \(D_i\) can separately or together verify and validate the signature by checking the following equations:

1. Computes \(a = (\bar{u})^{\prod_{i=1}^{5} e_{A_i}} \pmod{n_A}\)

2. Computes \(b = t^{\prod_{i=1}^{7} d_{B_i}} \pmod{n_B}\)

3. Check \(c = g_A^a (y_{A_1}y_{A_2}y_{A_3}y_{A_4}y_{A_5})^{-b} \pmod{p} = r^r \prod_{i=1}^{7} z_i \pmod{p}\)

4. Lastly, signatures are accepted if \(b = H_2(M, c)\).

5. If any of the condition from above two conditions fail, verifier can deny to accept the signature. In case of more than 2 verifiers, if 50% or more verifiers deny the signature, system \(B\) will return the signature to system \(A\).

**3.3 Algorithm 3**

**Set up**

Let \(K\) be the security parameter and \(sp = (E, P, Q, p, n_A, n_B, H_3) \leftarrow Setup(K)\), where

- \(p\) be a large prime,
- \(E\) be an elliptic curve over \(Z_p\) such that \(n_A\) and \(n_B\) are factors of \(|E|\), where \(n_A\) and \(n_B\) are product of two large primes,
• $P, Q \in E$ such that $|P| = n_A, |Q| = n_B$,

• $H_3$ be a cryptographically secured hash function with arbitrary bit length.

Key Generation

Key generation algorithm for system $A$:
1. System $A$ chooses prime $p_A$ and $q_A$, then computes $n_A = p_A \times q_A$.
2. Each $S_i$ follow the following steps:
   • randomly choose $e_{A_i}$ such that $\text{g.c.d.}(e_{A_i}, \phi(n_A)) = 1$.
   • compute $d_{A_i}$ such that $e_{A_i} d_{A_i} = 1 \pmod{\phi(n_A)}$.
   • choose an integer $x_{A_i} \in \mathbb{Z}_p^*$.
   • calculate $y_{A_i} = [x_{A_i}]P \pmod{p}$.
   • public key of $S_i$ is $(e_{A_i}, y_{A_i})$.
   • private key of $S_i$ is $(d_{A_i}, x_{A_i})$.
3. Note: Since, each member in system $A$ knows $\phi(n_A)$ and $e_{A_i} \ \forall i$, they can compute $d_{A_i}$. Thus, $d_{A_i}$ is private for members of system $B$ but not for members of system $A$.

Key generation algorithm for system $B$:
1. System $B$ chooses prime $p_B$ and $q_B$, then computes $n_B = p_B \times q_B$.
2. Each $D_i$ follow the following steps:
   • randomly choose $e_{B_i}$ such that $\text{g.c.d.}(e_{B_i}, \phi(n_B)) = 1$.
   • compute $d_{B_i}$ such that $e_{B_i} d_{B_i} = 1 \pmod{\phi(n_B)}$.
   • choose an integer $x_{B_i} \in \mathbb{Z}_p^*$.
   • calculate $y_{B_i} = [x_{B_i}]Q \pmod{p}$.
   • public key of $D_i$ is $(e_{B_i}, y_{B_i})$.
   • private key of $D_i$ is $(d_{B_i}, x_{B_i})$.
3. Note: Since, each member in system $B$ knows $\phi(n_B)$ and $e_{B_i} \ \forall i$, they can compute $d_{B_i}$. Thus, $d_{B_i}$ is private for members of system $A$ but not for members of system $B$.

Signature Algorithm

System $A$ generates a signature $(r, s, t, \bar{u})$ for a message $M$ as follows:
1. Each $S_i$ chooses a random integer $k_i$ and keep it secret.
2. Each $S_i$ generates $r_i = [k_i]P - [k_i](y_{B_1} + y_{B_2} + \ldots + y_{B_m}) \pmod{p}$, $s_i = [k_i]Q \pmod{p}$ and $w_i = [k_i]P \pmod{p}$ and make it public to the system $A$.
3. System computes:
\[ r = \sum_{i=1}^{n} r_i \pmod{p} \]

\[ s = \sum_{i=1}^{n} s_i \pmod{p} \]

\[ w = \sum_{i=1}^{n} w_i \pmod{p} \]

\[ z = H_3(M, w) \]

\[ t = z^{m} \prod_{i=1}^{m} e_{B_i} \pmod{n_B} \]

4. After computing \((r, s, w, z, t)\), they are made public to all \(S_i\) and then each \(S_i\) computes \(v_i = z \cdot x_{A_i} + k_i\) and make it public to the system \(A\).

5. System computes

\[ \bar{v} = \sum_{i=1}^{n} v_i \pmod{n_A} \]

\[ \bar{u} = (\bar{v})^{\prod_{i=1}^{n} d_{A_i}} \pmod{n_A} \]

System \(A\) sends the message \(M\) and the signature \((r, s, t, \bar{u})\) to the system \(B\) (and all the verifiers \(D_i\)).

**Verification Algorithm**

Upon receiving the message \(M\) and the signature \((r, s, t, \bar{u})\), each verifier \(D_i\) computes \(z_i = [x_{B_i}]s\) and make it public. Then, each \(D_i\) can separately or together verify and validate the signature by checking the following equations:

1. Computes \(a = (\bar{u})^{\prod_{i=1}^{n} e_{A_i}} \pmod{n_A}\)

2. Computes \(b = t^{\prod_{i=1}^{m} d_{B_i}} \pmod{n_B}\)

3. Check \(c = [a]P - b(y_{A_1} + y_{A_2} + ... + y_{A_n}) \pmod{p} = r + \sum_{i=1}^{m} z_i \pmod{p}\)

4. Lastly, signatures are accepted if \(b = H_3(M, c)\).

5. If any of the condition from above two conditions fail, verifier can deny to accept the signature. In case of more than 2 verifiers, if 50% or more verifiers deny the signature, system \(B\) will return the signature to system \(A\).
Transcript Simulation

System $B$ (or designated verifier $D$) can simulate the correct transcripts as follows:

1. System $B$ chooses a random integer $k'$ and keep it secret.

2. Then $B$ generates:
   - $r' = [k']Q - [k'](y_{A_1} + y_{A_2} + \ldots + y_{A_n}) \pmod{p}$
   - $s' = [k']P \pmod{p}$
   - $w' = [k']Q \pmod{p}$
   - $z' = H(m, w')$
   - $t' = z' \left( \prod_{i=1}^{n} e_{A_i} \right) \pmod{n_A}$
   - $v' = \sum_{i=1}^{m} v_i' \pmod{n_B}$, where $v_i' = z'x_{B_i} + k'$ is made public by $D_i$
   - $u' = (v')^{d_B} \pmod{n_B}$, where $d_B = \prod_{i=1}^{m} d_{B_i}$

System $B$ simulated signature is $(r', s', t', u')$.

Example 1

Let $A = \{S_1, S_2, S_3\}$ be the group of three signers (or provers) and $B = \{D_1, D_2\}$ be the group of two designated verifiers.

Let $p = 419$ be a prime and $E$ be an elliptic curve over $\mathbb{Z}_p$ such that

$$E = E_{\mathbb{Z}_p} = \{(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p / y^2 = x^3 + 2\}$$

Note that $|E| = 420$ and $E$ is a cyclic group. Also $n_A = 15$ and $n_B = 14$ are factors of $|E|$. Let $P = (22, 151)$, $Q = (55, 156)$ in $E$ such that $|P| = 15$, $|Q| = 14$.

Let $H_3$ be a cryptographically secured hash function with arbitrary bit length.

**Key generation algorithm for system $A$:**

1. System $A$ chooses prime $p_A = 3$ and $q_A = 5$, then computes $n_A = 3 \ast 5$.

2. Each $S_i$ follow the following steps :

   | $S_i$ | $e_{A_i}$ | $d_{A_i}$ | $x_{A_i}$ | $y_{A_i}$ |
   |-------|----------|----------|----------|----------|
   | $S_1$ | 13       | 5        | 7        | $7P$     |
   | $S_2$ | 7        | 7        | 16       | $P$      |
   | $S_3$ | 11       | 3        | 21       | $6P$     |

3. Public key of $S_i$ is $(e_{A_i}, y_{A_i})$.

4. Private key of $S_i$ is $(d_{A_i}, x_{A_i})$. 
5. Note: Since, each member in system $A$ knows $\phi(n_A)$ and $e_{A_i} \ \forall i$, they can compute $d_{A_i}$. Thus, $d_{A_i}$ is private for members of system $B$ but not for members of system $A$.

**Key generation algorithm for system $B$:**

1. System $B$ chooses prime $p_B = 2$ and $q_B = 7$, then computes $n_B = 2 \times 7$.

2. Each $D_i$ follow the following steps:

   | $D_i$ | $e_{B_i}$ | $d_{B_i}$ | $x_{B_i}$ | $y_{B_i}$ |
   |-------|-----------|-----------|-----------|-----------|
   | $D_1$ | 5         | 5         | 19        | 5$Q$      |
   | $D_2$ | 11        | 5         | 17        | 3$Q$      |

3. public key of $D_i$ is $(e_{B_i}, y_{B_i})$.

4. private key of $D_i$ is $(d_{B_i}, x_{B_i})$.

5. Note: Since, each member in system $B$ knows $\phi(n_B)$ and $e_{B_i} \ \forall i$, they can compute $d_{B_i}$. Thus, $d_{B_i}$ is private for members of system $A$ but not for members of system $B$.

**Signature Algorithm**

1. Each $S_i$ chooses a random integer $k_i$ and keep it secret and then, generates

   $r_i = [k_i]P - [k_i](y_{B_1} + y_{B_2} + \ldots + y_{B_e}) \quad \text{(mod $p$)}$ ,

   $s_i = [k_i]Q \quad \text{(mod $p$)}$ and

   $w_i = [k_i]P \quad \text{(mod $p$)}$

   and make it public to the system $A$:

   | $S_i$ | $k_i$ | $r_i$ | $s_i$ | $w_i$ |
   |-------|-------|-------|-------|-------|
   | $S_1$ | 8     | 8$P - 8Q$ | 8$Q$ | 8$P$ |
   | $S_2$ | 12    | 12$P - 12Q$ | 12$Q$ | 12$P$ |
   | $S_3$ | 14    | 14$P$ | 14$Q$ | 14$P$ |

2. System computes:

   - $r = \sum_{i=1}^{3} r_i \equiv 4P - 6Q \quad \text{(mod $p$)}$
   - $s = \sum_{i=1}^{3} s_i \equiv 6Q \quad \text{(mod $p$)}$
   - $w = \prod_{i=1}^{3} w_i \equiv 4P \quad \text{(mod $p$)}$
   - $z = H_3(M, w)$
   - $t = z^{i=1} \equiv z^{55} \equiv z \quad \text{(mod $14$)}$
3. After computing \((r, s, w, z, t)\), they are made public to all \(S_i\) and then each \(S_i\) computes \(v_i = zx_{A_i} + k_i\) and make it public to the system \(A\).

\[
v_1 = 7z + 8, \ v_2 = 16z + 12, \ v_3 = 21z + 14
\]

4. System computes

- \(\bar{v} = \sum_{i=1}^{3} v_i \equiv 14z + 4 \pmod{15}\)
- \(\bar{u} = (\bar{v})^3 \equiv \bar{v} \pmod{15}\)

System \(A\) sends the message \(M\) and signature \((r, s, t, \bar{u})\) to the system \(B\) (and all the verifiers \(D_i\)).

**Verification Algorithm**

Upon receiving the message \(M\) and the signature \((r, s, t, \bar{u})\), each verifier \(D_i\) computes \(z_i = [x_{B_i}]s\) and make it public. Then, each verifier \(D_i\) can separately or together verify and validate the signature by checking the following equations:

1. Computes \(a = (\bar{u})^{d_{A_i}} \equiv (\bar{v})^{5 \times 7 \times 3} \equiv \bar{v} \pmod{15}\)

2. Computes \(b = t^{d_{B_i}} \pmod{n_B}\)

3. Check \(c = [a]P - b(y_{A_1} + y_{A_2} + y_{A_3}) \pmod{p} = r + \sum_{i=1}^{2} z_i \pmod{p}\)

4. Lastly, signatures are accepted if \(b = H_3(M, c)\).

5. If any of the condition from above two conditions fail, verifier can deny to accept the signature. In case of more than 2 verifiers, if 50% or more verifiers deny the signature, system \(B\) will return the signature to system \(A\).

**Example 2**

Let \(A = \{S_1, S_2, S_3, S_4, S_5\}\) be the group of five signers (or provers) and \(B = \{D_1, D_2, D_3, D_4, D_5, D_6, D_7\}\) be the group of seven designated verifiers.

Let \(p = 6793\) be a prime and \(E\) be an elliptic curve over \(\mathbb{Z}_p\) such that

\[E = E_{\mathbb{Z}_p} = \{(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p / y^2 = x^3 + 5\}\]

Note that \(|E| = 6916\) and \(E\) is a cyclic group. Also \(n_A = 91\) and \(n_B = 38\) are factors of \(|E|\).

Let \(P = (3245, 4097)\), \(Q = (5223, 4702)\) in \(E\) such that \(|P| = 91\), \(|Q| = 38\).

Let \(H_3\) be a cryptographically secured hash function with arbitrary bit length.

**Key generation algorithm for system \(A\):**
1. System $A$ chooses prime $p_A = 7$ and $q_A = 13$, then computes $n_A = 7 \times 13$.

2. Each $S_i$ follow the following steps:

| $S_i$ | $e_{A_i}$ | $d_{A_i}$ | $x_{A_i}$ | $y_{A_i}$ |
|-------|-----------|-----------|-----------|-----------|
| $S_1$ | 5         | 29        | 15        | $15P$     |
| $S_2$ | 11        | 59        | 19        | $19P$     |
| $S_3$ | 7         | 31        | 24        | $24P$     |
| $S_4$ | 23        | 47        | 18        | $18P$     |
| $S_5$ | 35        | 35        | 32        | $32P$     |

3. public key of $S_i$ is $(e_{A_i}, y_{A_i})$.

4. private key of $S_i$ is $(d_{A_i}, x_{A_i})$.

5. Note: Since, each member in system $A$ knows $\phi(n_A)$ and $e_{A_i}$ $\forall i$, they can compute $d_{A_i}$. Thus, $d_{A_i}$ is private for members of system $B$ but not for members of system $A$.

Key generation algorithm for system $B$:

1. System $B$ chooses prime $p_B = 2$ and $q_B = 19$, then computes $n_B = 2 \times 19$.

2. Each $D_i$ follow the following steps:

| $D_i$ | $e_{B_i}$ | $d_{B_i}$ | $x_{B_i}$ | $y_{B_i}$ |
|-------|-----------|-----------|-----------|-----------|
| $D_1$ | 5         | 11        | 32        | $32Q$     |
| $D_2$ | 7         | 13        | 17        | $17Q$     |
| $D_3$ | 11        | 5         | 22        | $22Q$     |
| $D_4$ | 13        | 7         | 27        | $27Q$     |
| $D_5$ | 17        | 17        | 18        | $18Q$     |
| $D_6$ | 13        | 7         | 21        | $21Q$     |
| $D_7$ | 7         | 13        | 51        | $51Q$     |

3. public key of $D_i$ is $(e_{B_i}, y_{B_i})$.

4. private key of $D_i$ is $(d_{B_i}, x_{B_i})$.

5. Note: Since, each member in system $B$ knows $\phi(n_B)$ and $e_{B_i}$ $\forall i$, they can compute $d_{B_i}$. Thus, $d_{B_i}$ is private for members of system $A$ but not for members of system $B$.

Signature Algorithm

1. Each $S_i$ chooses a random integer $k_i$ and keep it secret and then, generates

$$r_i = [k_i]P - [k_i](y_{B_1} + y_{B_2} + \ldots + y_{B_e}) \pmod{p}$$

and make it public to the system $A$:
2. System computes:

- \( r = \sum_{i=1}^{5} r_i \equiv 19P - 36Q \pmod{p} \)
- \( s = \sum_{i=1}^{5} s_i \equiv 20Q \pmod{p} \)
- \( w = \prod_{i=1}^{5} w_i \equiv 19P \pmod{p} \)
- \( z = H_3(M, w) \)
- \( t = z \left( \prod_{i=1}^{7} e_{B_i} \right) \pmod{38} \)

3. After computing \((r, s, w, z, t)\), they are made public to all \( S_i \) and then each \( S_i \) computes \( v_i = zw_{A_i} + k_i \) and make it public to the system \( A \).

\[
\begin{align*}
v_1 &= 15z + 5770, \\
v_2 &= 19z + 2769, \\
v_3 &= 24z + 6476, \\
v_4 &= 18z + 1751, \\
v_5 &= 32z + 88
\end{align*}
\]

4. System computes

- \( \bar{v} = \sum_{i=1}^{5} v_i \equiv 17z + 19 \pmod{91} \)
- \( \bar{u} = (\bar{v}) \left( \prod_{i=1}^{5} d_{A_i} \right) \pmod{91} \)

System \( A \) sends the message \( M \) and the signature \((r, s, t, \bar{u})\) to the system \( B \) (and all the verifiers \( D_i \)).

**Verification Algorithm**

Upon receiving the message \( M \) and the signature \((r, s, t, \bar{u})\), each verifier \( D_i \) computes \( z_i = [x_{B_i}]s \) and make it public. Then, each \( D_i \) can separately or together verify and validate the signature by checking the following equations:

\[ \prod_{i=1}^{5} e_{A_i} \pmod{91} \]

1. Computes \( a = (\bar{u}) \pmod{91} \)
2. Computes \( b = \prod_{i=1}^{7} d_{B_i} \) \( (mod\ 38) \)

3. Check \( c = [a]P - b(y_{A_1} + y_{A_2} + y_{A_3} + y_{A_4} + y_{A_5}) \) \( (mod\ p) = r + \sum_{i=1}^{7} z_i \) \( (mod\ p) \)

4. Lastly, signatures are accepted if \( b = H_3(M, c) \).

5. If any of the condition from above two conditions fail, verifier can deny to accept the signature. In case of more than 2 verifiers, if 50% or more verifiers deny the signature, system \( B \) will return the signature to system \( A \).

3.4 Final Algorithm

In this section, we have combined algorithm 1 and 2 to enhance the security of the algorithm.

Set up

Let \( K \) be the security parameter and \( sp = (G, G_{\tau}, \tilde{G}, g, g_A, g_B, p, e, n_A, n_B, H_1, H_2) \gets \text{Setup}(K) \), where

- \( p \) be a large prime such that \( n_A \) and \( n_B \) are factors of \( p - 1 \),
- \( G \) is an additive cyclic group of large prime order \( p \) and \( g \) is a generator of \( G \),
- \( G_{\tau} \) is a multiplicative cyclic group of prime order \( p \),
- \( \tilde{G} = Z_p^* \) and \( g_A, g_B \in \tilde{G} \) such that \( |g_A| = n_A, |g_B| = n_B \),
- \( e : G \times G \to G_{\tau} \) is symmetric admissible bilinear pairing map,
- \( H_1 : \{0,1\}^* \to G \) is a cryptographically secure hash function,
- \( H_2 \) be a cryptographically secured hash function with arbitrary bit length.

Key Generation

**Key generation algorithm for system \( A \):**

1. System \( A \) chooses prime \( p_A \) and \( q_A \), then computes \( n_A = p_A * q_A \).
2. Each \( S_i \) follow the following steps:

   - choose \( a_i \in G \).
   - calculate \( p_i = a_i g \) \( (mod\ p) \).
   - randomly choose \( e_{A_i} \) such that \( \text{g.c.d.}(e_{A_i}, \phi(n_A)) = 1 \).
   - compute \( d_{A_i} \) such that \( e_{A_i} d_{A_i} = 1 \) \( (mod\ \phi(n_A)) \).
   - choose an integer \( x_{A_i} \in Z_p^* \).
   - calculate \( y_{A_i} = g_A^{x_{A_i}} \) \( (mod\ p) \).
• public key of $S_i$ is $(p_i, e_{A_i}, y_{A_i})$.
• private key of $S_i$ is $(a_i, d_{A_i}, x_{A_i})$.

3. System $A$, then, publishes $u = \sum_{i=1}^{n} p_i$.

4. Note: Since, each member in system $A$ knows $\phi(n_A)$ and $e_{A_i}$ $\forall i$, they can compute $d_{A_i}$. Thus, $d_{A_i}$ is private for members of system $B$ but not for members of system $A$. Similarly, $p_i$ is public to system $A$ only.

Key generation algorithm for system $B$:

1. System $B$ chooses prime $p_B$ and $q_B$, then computes $n_B = p_B * q_B$.

2. Each $D_i$ follow the following steps:
   - choose $b_i \in G$.
   - calculate $q_i = b_i g (mod p)$.
   - randomly choose $e_{B_i}$ such that $g.c.d.(e_{B_i}, \phi(n_B)) = 1$.
   - compute $d_{B_i}$ such that $e_{B_i} d_{B_i} = 1 (mod \phi(n_B))$.
   - choose an integer $x_{B_i} \in Z_p^*$.
   - calculate $y_{B_i} = g^{x_{B_i}} (mod p)$.
   - public key of $D_i$ is $(q_i, e_{B_i}, y_{B_i})$.
   - private key of $D_i$ is $(b_i, d_{B_i}, x_{B_i})$.

3. System $B$, then, publishes $v = \sum_{i=1}^{m} q_i$.

4. Note: Since, each member in system $B$ knows $\phi(n_B)$ and $e_{B_i}$ $\forall i$, they can compute $d_{B_i}$. Thus, $d_{B_i}$ is private for members of system $A$ but not for members of system $B$. Similarly, $q_i$ is public to system $B$ only.

Signature Algorithm

System $A$ generates a signature $(\sigma, r, s, t, \bar{u})$ for a message $M$ as follows:

1. Each $S_i$ chooses a random integer $k_i$ and keep it secret.

2. Each $S_i$ generates $\sigma_i = e(H_1(M), a_i v) \in G_\tau$, $r_i = g_A^{k_i} (y_{B_1} y_{B_2} \ldots y_{B_m})^{-k_i} (mod p)$, $s_i = g_B^{k_i} (mod p)$ and $w_i = g_A^{k_i} (mod p)$ and make it public to the system $A$.

3. System computes:
   - $\sigma = \prod_{i=1}^{n} \sigma_i \in G_\tau$
   - $r = \prod_{i=1}^{n} r_i (mod p)$
\[ s = \left( \prod_{i=1}^{n} s_i \right)^r \pmod{p} \]

\[ w = \left( \prod_{i=1}^{n} w_i \right)^r \pmod{p} \]

\[ z = H_2(M, w) \]

\[ t = z^{\prod_{i=1}^{m} e_{B_i}} \pmod{n_B} \]

4. After computing \((\sigma, r, s, w, z, t)\), they are made public to all \(S_i\) and then each \(S_i\) computes \(v_i = zx_{A_i} + k_ir\) and make it public to the system \(A\).

5. System computes

\[ \bar{v} = \sum_{i=1}^{n} v_i \pmod{n_A} \]

\[ \bar{u} = (\bar{v})^{\prod_{i=1}^{n} e_{A_i}} \pmod{n_A} \]

System \(A\) sends the message \(M\) and the signature \((\sigma, r, s, t, \bar{u})\) to the system \(B\) (and all the designated verifiers \(D_i\)).

**Verification Algorithm**

Upon receiving the message \(M\) and the signature \((\sigma, r, s, t, \bar{u})\), each designated verifier \(D_i\) computes \(\zeta_i = e(H_1(M), b_iu) \in G_{\tau}\) and \(z_i = s^{x_{A_i}} \pmod{p}\) and make it public. Then, each \(D_i\) can separately or together verify and validate the signature by checking the following equations:

1. Computes \(\zeta = \prod_{i=1}^{m} \zeta_i \in G_{\tau}\).

2. Computes \(a = (\bar{u})^{\prod_{i=1}^{n} e_{A_i}} \pmod{n_A}\)

3. Computes \(b = t^{\prod_{i=1}^{m} d_{B_i}} \pmod{n_B}\)

4. Check \(\sigma \equiv \zeta \pmod{G_{\tau}}\).

5. Check \(c = g_A^{x_A} (y_{A_1}y_{A_2}...y_{A_n})^{-b} \pmod{p} = r^r \prod_{i=1}^{m} z_i \pmod{p}\)

6. Lastly, signatures are accepted if \(b = H_2(M, c)\) and \(\sigma \equiv \zeta \pmod{G_{\tau}}\).

7. If any of the condition from above conditions fail, verifier can deny to accept the signature. In case of more than 2 verifiers, if 50% or more verifiers deny the signature, system \(B\) will return the signature to system \(A\).
4 Security Analysis

Now, we show the security properties for this scheme:

**Theorem 4.1.** Our MSMV-SDVS scheme is correct if it runs smoothly.

**Proof.** The signature verification procedure is correct since:

1. \[ \sigma \equiv \prod_{i=1}^{n} \sigma_i \equiv \prod_{i=1}^{n} e(H_1(M), a_i v) \equiv e(H_1(M), \sum_{i=1}^{n} a_i v) \equiv e(H_1(M), \sum_{i=1}^{n} \sum_{j=1}^{m} a_i b_j g) \]
   \[ \equiv e(H_1(M), \sum_{j=1}^{m} b_j u) \equiv \prod_{j=1}^{m} e(H_1(M), b_j u) \equiv \prod_{j=1}^{m} \zeta_j \equiv \zeta \text{ in } G_\tau \]

2. \[ a \equiv (\bar{u})^{i=1} \equiv (\bar{v})^{i=1} \equiv \bar{v} \text{ (mod } n_A) \]

3. \[ b \equiv t^{i=1} \equiv z^{i=1} \equiv z \text{ (mod } n_B) \]

4. \[ g_A^a (y_{A_1} \ldots y_{A_n})^{-b} \equiv g_A^a (g_A^{x_{A_1}} \ldots g_A^{x_{A_n}})^{-b} \equiv g_A^\bar{u} (g_A^{-b} \sum_{i=1}^{n} x_{A_i}) \equiv g_A^{r \sum_{i=1}^{n} k_i} \text{ (mod } p) \]

5. \[ r^r \prod_{i=1}^{m} z_i \equiv \prod_{i=1}^{m} s^{x_{B_i}} \equiv (\prod_{j=1}^{n} r_j)^r \sum_{i=1}^{m} x_{B_i} \]
   \[ \equiv (g_A^{z_1} (y_{B_1} \ldots Y_{B_m})^{-b} \sum_{j=1}^{n} k_j \sum_{i=1}^{m} x_{B_i} \equiv g_A^{r \sum_{j=1}^{n} k_j} \text{ (mod } p) \]

**Theorem 4.2.** Our MSMV-SDVS scheme is strong designated verifier signature scheme.

**Proof.** We have seen that the simulated signature produced by system B (or designated verifier D) is indistinguishable from the signature generated by system A. Also, the probability that the simulated signature produced randomly from the set of system A’s signature depends upon the randomness of \( k' \in \mathbb{Z}_{n_B}^* \) and it is \( 1/n_B \). Thus the two signatures have the same probability distribution, and hence the proposed scheme is a designated verifier signature scheme.

Our scheme involves each designated verifier and their secret keys \( b_i, x_{B_i} \) in the verification process. Moreover, the signature \( (\sigma, r, s, t, \bar{u}) \) produced by system A is indistinguishable from the signature \( (\zeta, r', s', t', \bar{u}') \) generated by system B. Thus, system B can not convince any third party whether a signature is created by signers or verifiers. Thus, our scheme satisfies the strongness property.

\[ \square \]
Theorem 4.3. Our MSMV-SDVS scheme is unforgeable unless an adversary has access to the secret keys of all signers and verifiers.

Proof. Any adversary, who does not know the private key of all signers and designated verifiers, can not forge the signature. It is computationally infeasible to forge the signature in polynomial time.

Notation: $1 \leq i \leq n$, $1 \leq j \leq m$, where $n$ and $m$ are number of signers and designated verifiers respectively.

- If attacker knows all $p_i$ or $q_j$, it is infeasible to compute all $a_i$ or $b_j$ in polynomial time.
- If attacker knows all $a_i$ or $b_j$, signature $\sigma$ or $\zeta$ can be created but to compute $r, s, t, \bar{u}$, he further require $d_{A_i}, x_{A_i}, d_{B_j}, x_{B_j}$ $\forall i, j$

Since our scheme is based on bilinear pairing, factorization problem, and discrete logarithm problem, it is computationally infeasible to compute $\sigma, r, s, t, \bar{u}$ in polynomial time.

Theorem 4.4. Our MSMV-SDVS scheme is non-transferable, and protects the identity of the actual signer.

Proof. In this scheme, signature $(\zeta, r', s', t', \bar{u}')$ produced by system B is indistinguishable from the signature $(\sigma, r, s, t, \bar{u})$ generated by system A. Thus, individually or collectively, Verifiers can not convince anyone that the signer generated the signature, and any third party can not determine who generated which signature. Thus our scheme is non-transferable and protects the identity of the actual signer.

5 Conclusion

This paper presents a multi-signer strong designated multi-verifier signature scheme based on multiple cryptographic algorithms. Since all the signers generate their signatures individually and make them public to the system to generate a common signature, this scheme is non-interactive. Also, our scheme requires the authorization of each participating signer, making it applicable to various dimensions using blockchain technology. Our scheme satisfies strongness, unforgeability, and non-transferability properties in the random oracle model.
References

[1] Ting-Yi Chang. An id-based multi-signer universal designated multi-verifier signature scheme. *Information and Computation*, 209(7):1007–1015, 2011.

[2] David Chaum and Hans Van Antwerpen. Undeniable signatures. In *Conference on the Theory and Application of Cryptology*, pages 212–216. Springer, 1989.

[3] Liao Da-jian and Tang Yuan-sheng. Comment on lee et al.’s strong designated verifier signature scheme and its improvement. In *Advances in Technology and Management*, pages 309–314. Springer, 2012.

[4] Lunzhi Deng, Jiwen Zeng, and Huawei Huang. Id-based multi-signer universal designated multi-verifier signature based on discrete logarithm. *Chiang MAI J. Sci*, 45(1):617–624, 2018.

[5] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[6] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[7] SK Hafizul Islam and GP Biswas. Certificateless strong designated verifier multisignature scheme using bilinear pairings. In *Proceedings of the international conference on advances in computing, communications and informatics*, pages 540–546, 2012.

[8] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated verifier proofs and their applications. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 143–154. Springer, 1996.

[9] Asif Uddin Khan and Bikram Kesari Ratha. A secure strong designated verifier signature scheme. *Int. J. Netw. Secur.*, 19(4):599–604, 2017.

[10] Fabien Laguillaumie and Damien Vergnaud. Multi-designated verifiers signatures. In *International Conference on Information and Communications Security*, pages 495–507. Springer, 2004.

[11] Fabien Laguillaumie and Damien Vergnaud. Multi-designated verifiers signatures: anonymity without encryption. *Information Processing Letters*, 102(2-3):127–132, 2007.

[12] Ji-Seon Lee and Jik Hyun Chang. Strong designated verifier signature scheme with message recovery. In *The 9th International Conference on Advanced Communication Technology*, volume 1, pages 801–803. IEEE, 2007.

[13] Ji-Seon Lee and Jik Hyun Chang. Comment on saeednia et al.’s strong designated verifier signature scheme. *Computer Standards & Interfaces*, 31(1):258–260, 2009.

[14] Xiangxue Li and Kefei Chen. Id-based multi-proxy signature, proxy multi-signature and multi-proxy multi-signature schemes from bilinear pairings. *Applied Mathematics and Computation*, 169(1):437–450, 2005.

[15] Yang Ming and Yumin Wang. Universal designated multi verifier signature scheme without random oracles. *Wuhan University Journal of Natural Sciences*, 13(6):685–691, 2008.
[16] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[17] Shahrokh Saeednia, Steve Kremer, and Olivier Markowitch. An efficient strong designated verifier signature scheme. In *International conference on information security and cryptology*, pages 40–54. Springer, 2003.

[18] Pankaj Sarde and Amitabh Banerjee. Strong designated verifier signature scheme based on discrete logarithm problem. *Journal of Discrete Mathematical Sciences and Cryptography*, 18(6):877–885, 2015.

[19] N Shapuan and ES Ismail. A new strong designated verifier signature scheme. In *AIP Conference Proceedings*, volume 1940, page 020122. AIP Publishing LLC, 2018.

[20] Nadiah Shapuan and Eddie Shahril Ismail. A strong designated verifier signature scheme with hybrid cryptographic hard problems. *Journal of Applied Security Research*, pages 1–13, 2021.

[21] Shiang-Feng Tzeng, Cheng-Ying Yang, and Min-Shiang Hwang. A nonrepudiable threshold multi-proxy multi-signature scheme with shared verification. *Future Generation Computer Systems*, 20(5):887–893, 2004.

[22] Guilin Wang. An attack on not-interactive designated verifier proofs for undeniable signatures. *IACR Cryptol. ePrint Arch.*, 2003:243, 2003.

[23] Bo Yang, Zibi Xiao, Yixian Yang, Zhengming Hu, and Xinxin Niu. A strong multi-designated verifiers signature scheme. *Frontiers of Electrical and Electronic Engineering in China*, 3(2):167–170, 2008.

[24] Yaling Zhang, Jing Zhang, and Yikun Zhang. Multi-signers strong designated verifier signature scheme. In *2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pages 324–328. IEEE, 2008.