Analysis of a Key-Escrow Free Multi-Signature Scheme

Mehbub Alam[†] and Subhas C. Sahana[‡]

[†]Department of Computer Science & Engineering, Indian Institute of Information Technology Guwahati, India
[‡]Department of Computer Science & Engineering, National Institute of Technology, Durgapur, India
E-mail: mehbub@iiitg.ac.in, subhaschandra.sahana@cse.nitdgp.ac.in

***Abstract*—The identity-based signature scheme needs no certificates as it has a flexible key management procedure. However, it has a significant drawback: the Key Generation Centre (KGC) knows the user's private keys, commonly known as the Key-Escrow problem. In this paper, we analyze an existing key-escrow free multi-signature scheme, proposed by Das *et al.* from bilinear pairings and blinding-binding technique. It is found that the sequential multi-signature scheme proposed by Das et al. is not actually following the characteristics of an ideal sequential multi-signature scheme and thus leads to the inefficiency of the scheme. Also, their parallel multi-signature scheme is not verifiable. We modified both schemes and designed an efficient new sequential multi-signature and a parallel multi-signature scheme.**

***Index Terms*—Bilinear pairing, Multi-signature, Sequential-multi-signature, Parallel multi-signature, Blinding binding technique.**

I. Introduction

The digital signature is a very important cryptographic primitive demonstrating the authenticity, integrity, and non-repudiation of a digital message or a digital document. A digital signature consists of a key generation algorithm, a signing algorithm, and a signature verifying algorithm. A user signs on a message using his private key, and the signature verification is done using the signer's public key. Digital signatures are standards of cryptographic protocol suit. In a traditional Public Key Infrastructure (PKI) [1] based cryptosystem, the authenticity of the public key is preserved by an issued certificate from a certificate authority. As a result, one has to verify the certificate to get a verified public key before encrypting a message or verifying a message's signature. Still, the inconvenience happens with the key management when the number of users becomes large and costly, along with certificate storage and revocation issue.

An identity-based signature scheme was proposed by Shamir [2] in 1984 with the goal of overcoming the difficulties in conventional PKI-based cryptosystems. Identity-based signature scheme [3] enables the user to get a public key without exchanging public key certificates. Users can generate their public using their identity. Shamir demonstrated the working system for identity-based signature (IDS), but there was no practical implementation till 2001. The First Identity Based Encryption scheme was discovered by Boneh and Franklin [4] in 2001 based on Weil pairing. In an identity-based signature scheme, Key Generator Centre (KGC) replaces Certificate Authority (CA), and the public key is some publicly known unique information like email-id, telephone no., employee id, etc. Identity-based cryptography is simpler, but it has two significant drawbacks: (a) the key-escrow problem and (b) the requirement of a secure channel for transmitting the generated private key from KGC to a user. Many identity-based signature schemes are introduced from bilinear pairing; however, all the schemes suffer from the same inherent problems. A few approaches [4]–[8] have been presented as solutions to those mentioned drawbacks, but they failed due to either computational cost overhead or communication cost overhead or both.

Multi-signature is a variant of digital signature where the multi-signature is generated from n signatures signed by n signers, where $(n > 1)$ is on a single message. The verification of the message can be done by any user. The concept of multi-signature was introduced by Itakura and Nakamura [9] in 1983. Multi-signature was divided into two types based on the application requirements. They are Sequential Multi-Signature (SMS) and Parallel Multi-Signature (PMS). In the SMS scheme, every signer signs the message sequentially, one after another in a predefined order. Here, the first signer needs to sign the message only, but the intermediate and end users must perform both verification and signing. The final output signature generated by the end signer is the multi-signature of the message. In the parallel multi-signature scheme, each signer creates their individual signature on the message. Lastly, a designated clerk combines each individual signature from each signer into a multi-signature after successfully validating the individual signatures.

The rest of the paper is organized as follows : some preliminaries have been discussed in section II. Section III briefly reviews the serial multi-signature scheme and parallel multi-signature scheme proposed by Das *et al.* [10]. In section IV, we have proposed a serial multi-signature scheme and a parallel multi-signature scheme after modifying and eliminating the loopholes in [10]. We concluded our work in section V.

II. Preliminaries

A. Bilinear Pairing:

It is an important cryptographic tool and is widely adopted in many positive applications of cryptographic primitives. Let G_1 and G_2 are additive and multiplicative cyclic groups respectively of prime order q with P as a generator of G_1. A

bilinear pairing is a map e defined by $e : G_1 \times G_1 \rightarrow G_2$ satisfying the following properties:

- Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$, for all $a, b \in Z_q$ and $P, Q \in G_1$.
- Non-degeneracy: There exists $P, Q \in G_1$, such that $e(P, Q) \neq 1$.
- Computability: There is an efficient algorithm to compute $e(P, Q)$, for all $P, Q \in G_1$

B. *Decision Diffie-Hellman Problem (DDHP)*

For $a, b, c \in Z_q^*$. If P, aP, bP, cP is given to decide whether $c \equiv ab \bmod q$ is computationally hard and is known as Decisional Diffie-Hellman Problem. The DDHP is solvable in polynomial time as bilinear pairing can be used as tool to solve this problem.

C. *Computational Diffie-Hellman Problem (CDHP):*

For given $a, b \in Z_q^*$, compute abP for given (a, aP, bP). The advantage for solving CDHP in G_1 using any probabilistic polynomial-time algorithm B is defined as $Adv_{(P,G_1)}^{CDHP} = Pr[B(P, aP, bP, abP) = 1]$ is negligible. It is assumed that the CDHP problem is a hard problem.

III. Brief review of a key-escrow free multi-signature scheme proposed by Das *et al.* [10]

M. L. Das claimed that his proposed scheme [10] is a multi-signature scheme based on bilinear pairings which is not only key-escrow free but also does not need any secure channel for transmission of private keys. The technique called blinding binding technique has been used in the scheme to overcome from key escrow problem and to avoid secure channel problem for private key issuance to user. In his work, a sequential as well as a parallel multi-signature scheme has been proposed from the basic scheme [11] using bilinear pairings.

A. Blinding-binding scheme

Three parties involved in the model, are as follows [12]:

1) Key Generator Centre (KGC): The trusted authority whose role is to furnish the user with partial private key. The master secret key is the s, chosen randomly from Z_q^* and the public key $PK = sP$.
2) Signer: Signs the message using the private key.
3) Verifier: Verifies the signature using the user's public key along with the message received.

Two secret parameters $x, y \in Z_q^*$ are chosen by the user with the identity uid and her public key is computed as $UPK = H(uid)$, where UPK is the public key and $H : \{0,1\}^* \rightarrow G_1$ is map-to-point function. Then four binding parameters X, Y, Z, W are computed as: $X = x.UPK, Y = x.y.UPK, Z = y.P$ and $W = x.y.P$, where P is the generator of G_1. She then sends $< X, Y, Z, W, uid >$ to the KGC over a public channel.

- After receiving data sent by the user, the KGC checks its directory to check if the uid already exists. The KGC then sends an email verification message to the user's email id if the uid does not exists in its directory to prevent an unregistered identity attack. As soon as, the confirmation is done the KGC calculates $UPK = H(uid)$ and checks the following condition $e(Y, P) = e(X, Z) = e(UPK_{ID}, W)$. If the condition is holds, the KGC calculates users partial key $D_u = s.Y$, user's registration status $U_{st} = s.Z$. and inserts the value $< U_{st}, uid >$ in a public directory, sending D_u to the user over the public channel. The algorithm is defined as $D_u \leftarrow blinding - binding(params, uid, X, Y, Z, W)$
- After receiving D_u from KGC, the user checks the condition $e(D_u, P) = e(Y, PK)$. If the condition is valid then the user generates the private key $USK = x^{-1}D_u = y.s.UPK$ after unbinding.

In [10], at first a key escrow free identity based signature scheme was proposed from Hess's identity-based signature scheme [12]. After that, the proposed key escrow free identity-based signature scheme (Basic Signature Scheme (BSS)) was further extended to a key escrow free sequential and parallel multi-signature schemes. The multi-signature allows a strong verification mechanism to check that the message is indeed signed by each designated signer. In the scheme some assumptions are made such that KGC maintains a list of the registered users to whom the partial key was already provided. It is also assumed that at least one signer must remain honest in the multi-signature scheme even if adversary can control other signers.

B. Review of the undertaken Basic Signature Scheme (BSSS) in [10]

The BSS has four polynomial-time algorithms:

Setup: This is a randomized system generation algorithm that has security parameters as input and generates certain parameters. The parameters include KGC's master's secret keys $\in Z_q^*$, KGC's public key $PK = sP$, an additive group G_1, a multiplicative group G_2 (both of prime order q), a generator P of G_1, a bilinear pairinge $G_1 \times G_1 \rightarrow G_2$, a map-to-point$H : \{0,1\}^* \rightarrow G_1$ and a cryptographic hash function $h : \{0,1\}^* \times \{0,1\}^* \rightarrow Z_q^*$.

UserKeyGen: In this step the user secret key USK and the user public key UPK is generated using the blinding factors a,b and the binding parameters W, X, Y, Z, and the KGC's secret key s.

Sign: The signature of the message m is created in this step as follows: A random secret value $k \in Z_q^*$ is selected.

1) Calculate $r = e(P, P)^k$
2) Calculate $c = h(m, r, U_{st})$ and $\sigma = c.USK + kP$

The signature is (σ, c) for the message m.

Verify: The signature of the message is verified using the user's public key UPK, public key status U_{st}, public parameters and c. The signature is accepted only if the following condition is satisfied: $c = h(m, r^{'})$,if

$r^{'} = e(\sigma, P).e(UPK, -U_{st})^{c}$

C. Review of the proposed Sequential Multi-signature (SMS) scheme in [10]

Using this method all the signers are allowed to sign a message in a pre-agreed sequence one at a time (depending on hierarchy or using some other measures). The sequential multi-signature scheme consists of four algorithms. They are **Setup**, **UserKeyGen**, **MSign** and **MVerify**. The Setup and UserKeyGen steps are the same as the basis signature scheme.

Let $ID_1, ID_2, ID_3...ID_n$ be the identity of the users $u_1, u_2, u_3, ...u_n$. For $i = 1, 2, 3...n$, for each user has the public key is $UPK_i = H(ID_i)$, the private key is $USK_i = y_i.s.UPK_i$ and registration $ID = U_{(i,st)}$

Multi-signature Generation: The message m is signed sequentially by n signers and the output of the n^{th} signer is the multi-signature and sent to the verifier. The first signer needs to perform only the signing operation but the intermediate signers and the n^{th} signer needs to perform both the signing and the verification algorithm. The sequence for signing and verification is fixed.

Signature generation (first signer): The first signer with the registration identity $U_{(1,st)}$ generates signature on the message m in the following way :
I. Pick $k_1 \in_R Z_q^*$.
II. Compute $r_1 = e(P, P)^{k_1}$, $c_1 = h(m, r_1, U_{(1,st)})$ and $\sigma_1 = c_1.USK_1 + k_1P$.
III. Send $(\sigma_1, c_1, m, U_{(1,st)})$ to the next signer. Signature generation and verification by i^{th} intermediate user and nth signer: The i^{th} signers where $i = 2, 3...n$, needs to first verify the signature received from the previous signer i.e.
from $(i-1)^{th}$ signer which is $(\sigma_{(i-1)}, c_1, c_2...c_{(i-1)}, m, U_{(1,st)}, U_{(2,st)}, ...U_{(i-1,st)})$ by performing the following:
I. Compute
$r'_{(i-1)} = e(\sigma_{(i-1)}, P).e(UPK_{(i-1)}, -U_{(n-1,st)})^{C_{(i-1)}})$.
II. The signature is accepted only if
$c_{(i-1)} = h(m, r^{'}_{(i-1)}, U_{(i-1,st)})$.

The i^{th} signer generates the signature for the next $(i+1)^{th}$ in the following way:
I. Pick $k_i \in_R Z_q^*$.
II. Compute $r_i = e(P, P)^{k_i}, c_i = h(m, r_i)U_{(i,st)})$, and $\sigma_i = c_i.USK_i + k_iP$.
III. Send the signature tuple $(\sigma_i, c_1, c_2..., c_i, m, U_{(1,st)}, U_{(2,st)}..., U_{(i,st)})$ to the $(i+1)^{th}$signer.

Signature generation by n^{th} signer: The n^{th} signer creates the signature in the following way:
I. Pick $k_i \in_R Z_q^*$.
II. Compute $r_n = e(P, P)^{k_n}$, $c_n = h(m, r_n, UPK_{(n,st)})$ and $\sigma_n = c_n.USK_{(n,st)} + k_nP$.
The final signature $(_n, c_1, c_2..., c_n, m, U_{(1,st)}, U_{(2,st)}..., U_{(n,st)})$ is send to the verifier.

Multi-signature verification: After receiving the final signature $(\sigma_n, c_1, c_2..., c_n, m, U_{(1,st)}, U_{(2,st)}..., U_n)$ the verifier does the following :
I. Computes $r'_n = e(\sigma_n, P).e(UPK_n, -U_{(n,st)})^{c_n}$
II. The signature is accepted only if $c_n = h(m, r^{'}_n, U_{(n,st)})$.

Remark: This is not a proper sequential multi-signature as for every signature except the first signature does not contain any component from the previous signature. So, each generated individual signature is independent each of other which does not follow the property of an ideal sequential multi-signature.

Review of the proposed parallel multi-signature scheme (PMS) in [10]

In the parallel multi-signature scheme (PMS), n signers having identity ID_i where $i = 1, 2, 3, ...n$ signs independently on a message m generating individual signatures. Here one of the signers is elected as a designated clerk (DC) who is responsible for collecting, verifying and combining all the individual signatures. The multi-signature in this scheme is generated in the following ways:

Multi-signature generation : For $i = 1, 2, 3...n$, i^{th} signer with the identity ID_i creates his/her individual signatures in the following way:
a. Pick $k_i \in_R Z_q^*$.
b. Compute $r_i = e(P, P)^{k_1}$ and each signer broadcasts the value of r_i to the remaining $(n-1)$ signers.
c. Each signer with ID_i computes $r = \prod_{i=1}^{n} r_1$ and $c_i = h(m, r, U_{(i,st)})$
d. Compute $\sigma_i = c_i.USK_i + k_iP$.
e.The signature $(\sigma_i, c_i, m, U_{(i,st)})$ for each signer with ID_i is send to the designated clerk.
f. After receiving the signature the clerk verifies the individual signature $(\sigma_i, c_i, m, U_{(i,st)})$ by checking if the following equation is valid :
i. $c_i = h(m, e(\sigma_i, P).e(c_i.UPK_i, -PK), U_{(i,st)})$.
After the successful validation of the individual signatures the clerk computes $\sigma = \sum_{(i=1)}^{n} \sigma_i$ and $c = \prod_{i=1}^{n} c_i$
g. The multi-signature is (σ, c, m), where m is the message.

Multi-signature verification : The verification of the multi-signature is done in the following way : I. The equality of the equation $r^{'} = \prod_{i=1}^{n} r_i = e(\sigma, P).e(\sum_{i=1}^{n} UPK_i, -PK)^{c_i}$ is checked first. II. The multi-signature is accepted if and only if $c = \prod_{i=1}^{n} c_i$ where $c_i = h(m, r^{'}, U_{(i,st)})$.

Remark: The above proposed parallel multi-signature scheme is not verifiable because KGC's public key

$PK = s.Pb_isP$ **, so we cannot get the value** $r^{'}$ **as given below.**

$r^{'} = \prod_{i=1}^{n} r_i = e(\sigma, P).e(\sum_{i=1}^{n} UPK_i, -PK)^{c_i}$

$= e(\sum_{i=1}^{n} \sigma_i, P)e(\sum_{i=1}^{n} UPK_i, -PK)^{c_i}$

$=e(\sum_{i=1}^{n} c_i.USK_i + k_iP, P)e(\sum_{i=1}^{n} UPK_i, -PK)^{c_i}$

$=e(\sum_{i=1}^{n} c_i.USK_i, P) \prod_{i=1}^{n} e(P, P)^{k_i} e(\sum_{i=1}^{n} UPK_i, -PK)^{c_i}$

$=e(\sum_{i=1}^{n} c_i.b_isUPK_i, P) \prod_{i=1}^{n} e(P, P)^{k_i} e(\sum_{i=1}^{n} UPK_i, -PK)^{c_i}$

$=e(\sum_{i=1}^{n} UPK_i, b_isP)^{c_i} \prod_{i=1}^{n} e(P, P)^{k_i} e(\sum_{i=1}^{n} UPK_i, -PK)^{c_i}$

$\neq e(\sum_{i=1}^{n} UPK_i, -PK)^{c_i} \prod_{i=1}^{n} e(P, P)^{k_i} e(\sum_{i=1}^{n} UPK_i, -PK)^{c_i}$

IV. PROPOSED MULTI-SIGNATURE SCHEME

After analyzing the proposed existing schemes (SMS and PMS), both the schemes have been modified accordingly and proposed new corrected schemes. In the existing SMS scheme, the multi- signature has been created by the n^{th} signer. All the individual signatures are independent each of other. The end user verified the previous signature which does not resembles sequential characteristics. It is also seen that the multi-signature created using the parallel multi-signature scheme is not verifiable. So, for solving those problems, we have introduced new multi-signature schemes (SMS and PMS) that overcome all the problems of the reviewed existing multi-signature scheme.

A. The Proposed Sequential Multi-signature Scheme

In our proposed scheme the first signer just generates a signature like the BSS but when each of the other signers creates a signature after verifying the previous received signature, it simply add the previous signature such that there exists a dependency between the each signer' signature and it is the property of a sequential multi-signature scheme. Afterwards, a multi-signature is generated as an output of the n^{th} signer. The proposed SMS given below:

- The Setup and UserKeyGen step are the same as the basis signature scheme given in section 3.1.
- For $i = 1, 2, 3, ...n$, each user u_i having identity ID_i, the public key is computed as $UPK_i = H(ID_i)$ and the private key is also computed as $USK_i = y_i.s.UPK_i$.

The multi-signature generation and multi-signature verification step are as follows

Multi-signature generation: The message m is signed sequentially by n signers. The first signer needs to perform only signing operation but the intermediate and the end signer must perform both verification and the signing operation. The sequence for signing and verification is fixed and is done in the following way :

Signature generation by first signer:
a. Pick $k_1 \in_R Z_q^*$.
b. Compute $r'_1 = e(P, P)^{k_1}$
c. $r_1 = r_1^{'}$
d. $c_1 = h(m, r_1)$ and $\sigma_1 = c_1.USK_1 + k_1P$.
Send the tuple $(\sigma_1, c_1, m, U_{(1,st)})$ to the next signer.

- **Signature generation and verification (i^{th}intermediate user):** For the intermediate users where $i = 2, 3...(n-1)$ the signer needs to first verify the signature received from the previous signer i.e. for i^{th} signer must verify the signature from $(i-1)^{th}$ signer which is $(\sigma_{(i-1)}, c_1, c_2...c_{(i-1)}, m, U_{(1,st)}, U_{(2,st)}, ...U_{(i-1,st)})$ by performing the following:

  $r'_{(i-1)} = e(\sigma_{i-1}, P) \prod_{j=1}^{i-1} e(c_j PK_j Reg_{IDj})$

  The signer is accepted if $c_{i-1} = h(m_{i-1}, r_{i-1})$.
  First signers signature is verified by second signer i.e. $i = 2$

- **The i^th signer takes a value** $k_1 \in_R Z_q^*$ and computes

The i^th signer takes a value $k_1 \in_R Z_q^*$ and computes
a. Pick $k_i \in_R Z_q^*$ and computes
b. Compute $r_1^{'} = e(P, P)^{k_1}$
c. $r_i = t_{(i-1)}r_i^{'}$
d. $c_i = h(m_i, r_i)$
e. $\sigma_i = \sigma_{i-1} + c_iSK_ID_i + k_iP$

Verification and signature generation $(n^{th}$ **user):**
The n^{th} signer verifies the signature send by the $(n-1)^{th}$ signer which is $(\sigma_{n-1}, c_1, c_2, ..., c_n, U_{(1,st,})U_{(2,st)}, ...U_{(n-1,st)})$ by performing the following:

1. Compute $r_{n-1} = e(\sigma_{n-1}, P).e \prod_{j=1}^{i-1} e(c_j PK_{IDj} Reg_{ID}j)$

2. Accepts the signature if and only if $c_{n-1} = h(m, r_{n-1}^{'}, U_{(i-1,st)})$.

The n^{th} signer creates the signature in the following way:

a. Pick $k_i \in_R Z_q^*$.
b. Compute $r_n = e(P, P)^{k_n}$,
c. $c_n = h(m, r_n, UPK_n)$ and $\sigma_n = c_n.USK_n + k_nP$.

The final signature $(\sigma_n, c_1, c_2 \ldots, c_n, m, U_{(1,st)}, U_{(2,st)}..., U_{(n,st)})$ is send to the verifier

Multi-signature verification: After receiving the final signature $(\sigma_n, c_1, c_2..., c_n, m, U_{(1,st)}, U_{(2,st)}..., U_{(n,st)})$ the verifier does the following:

a.Computes $r'_n = e(\sigma_n, P).e\prod_{j=1}^{i-1} e(c_i PK_{IDi} Reg_{IDi})$
b. The signature is accepted only if $c_n = h(m, r'_n, U_{(n,st)})$

Correctness

$= e(\sum_{i=1}^{n}(c_i S_{IDi}) + k_i P)P) \prod_{i=1}^{n} e(c_i PK_{IDi} - Reg_{IDi})$

$= e(\sum_{i=1}^{n}(c_i SK_{IDi}), P)(\sum_{i=1}^{n} k_i P, P)$
$\prod_{i=1}^{n} e(c_i PK_{IDi} - Reg_{IDi})$

$= (\sum_{i=1}^{n} e(c_i sy_{IDi} PK_{IDi}.P))(\sum_{i=1}^{n} k_i P, P)$
$\prod_{i=1}^{n} e(c_i PK_{IDi} - Reg_{IDi})$

$= \prod_{i=1}^{n} e(c_i PK_{IDi}, sy_{IDi} P) e(\sum_{i=1}^{n} k_i P, P)$
$\prod_{i=1}^{n} e(c_i PK_{IDi} - Reg_{IDi})$

$= \prod_{i=1}^{n} e(c_i PK_{IDi} Reg_{IDi}) e(\sum_{i=1}^{n} k_i P, P)$
$\prod_{i=1}^{n} e(c_i PK_{IDi} - Reg_{IDi})$

$= (\sum_{i=1}^{n} k_i P, P)$

$= r_n$

Hence it has been proved that above signature scheme is correct

B. The Proposed Parallel Multi-signature Scheme

The Setup and UserKeyGen step are the same as the basis signature scheme given in section 3.1.
For $i = 1, 2, 3, ...n$, each user u_i having identity ID_i, the public key is computed as $UPK_i = H(ID_i)$ and the private key is also computed as $USK_i = y_i.s.UPK_i$.

Multi-signature generation : For $i = 1, 2, 3...n, i^{th} signer with the identity$ ID$_i$, creates his/her individual signatures in the following way:

a. Pick $k_i \in_R Z_q^*$.
b. Compute $r_i = e(P, P)^{k_1}$ and each signer broadcasts the value of r_i to the remaining (n-1) signers
c. Each signer with ID_i computes $r = \sum_{i=1}^{n} r_i$ and $c_i = h(m, r, U_{(i,st)})$
d. Compute $\sigma_i = c.USK_i + k_i P$.
e. The signature $(\sigma_i, c_i, m, U_{(i,st)})$ for each signer with ID_i is send to the designated clerk.
f. After receiving the signature the clerk verifies the individual signature $(\sigma_i, c_i, m, U_{(i,st)})$ by checking if the following equation is valid :
$c_i = h(m, e(\sigma_i, P).e(c_i.UPK_i, -Reg_{IDi}), U_{(i,st)})$

After the successful validation of the individual signatures the clerk computes
$\sigma = \sum_{i=1}^{n} \sigma_i$ and $c = \prod_{i=1|}^{n} c_i$

g. The multi-signature is (σ, c, m), where m is the message.

Multi-signature verification : The verification of the multi-signature is done in the following way :

a. The equality of the equation
$r' = \prod_{i=1}^{n} r_i = e(\sigma, P).e(\sum_{i=1}^{n} UPK_i, -Reg_{ID_i})^{c_i}$
is checked first.

b. The multi-signature is accepted if and only if $c = \prod_{i=1}^{n} c_i$ where $c_i = h(m, r', U_{(i,st)})$

Correctness

$e(\sigma, P) \prod_{i=1}^{n} e(cPK_{IDi} - Reg_{IDi})$

$= e(\sum_{i=1}^{n} \sigma_i P) \prod_{j=1}^{i-1} e(cPK_{IDi} - Reg_{IDi})$

$= e(\sum_{i=1}^{n}(c_i SK_{IDi} + k_i P) \prod_{i=1}^{n} e(cPK_{ID}i - Reg_{IDi})$

$= e(\sum_{i=1}^{n}(c_i SK_{IDi}, P), e(\sum_{i=1}^{n} k_i P, P) \prod_{i=1}^{n} e(cPK_{IDi} -$
$Reg_{IDi})$

$= \prod_{i=1}^{n} e(c_i sy_{IDi}, PK_{IDi}, P) e(\sum_{i=1}^{n} k_i P, P) \prod_{i=1}^{n} e(cPK_{IDi} -$
$Reg_{IDi})$

$= \prod_{i=1}^{n} e(c_i PK_{IDi}, sy_{IDi}, P) e(\sum_{i=1}^{n} k_i P, P) \prod_{i=1}^{n} e(cPK_{IDi} -$
$Reg_{IDi})$

$= \prod_{i=1}^{n} e(cPK_{IDi} Reg_{IDi}) e(\sum_{i=1}^{n} k_i P, P)$
$\prod_{i=1}^{n} e(PK_{IDi} - Reg_{IDi})$

$= e(\sum_{i=1}^{n} k_i P, P)$

$= \prod_{i=1}^{n} e(k_i P, P)$

$= \prod_{i=1}^{n} r_i$
$= r$

V. CONCLUSION

Our whole work is divided into two parts. First we analyze two types of existing proposed key escrow free multi-signature schemes (SMS and PMS) and found that the existing sequential multi-signature scheme is not a proper sequential multi-signature scheme as for every signature except the first signature does not contain any component from the previous signature. So, each generated individual signature is independent each of other which violets the property of an ideal sequential multi-signature. Moreover, the existing parallel Multi-signature scheme is not verifiable. After finding out those problems, we redesign both the schemes and propose new SMS and PMS schemes and claim that our proposed schemes (both SMS and PMS) remove all the drawbacks in the existing schemes.

REFERENCES

[1] A. Salomaa, *Public-Key Cryptography*. Monographs in Theoretical Computer Science. An EATCS Series, Springer Berlin Heidelberg, 2013.

[2] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Conference on the theory and application of cryptographic techniques*, pp. 186–194, Springer, 1986.

[3] P. S. Barreto, "The pairing-based crypto lounge," *http://www. larc. usp. br/pbarreto/pblounge. html*, 2005.

[4] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Annual international cryptology conference*, pp. 213–229, Springer, 2001.

[5] L. Chen, K. Harrison, D. Soldera, and N. P. Smart, "Applications of multiple trust authorities in pairing based cryptosystems," in *International Conference on Infrastructure Security*, pp. 260–275, Springer, 2002.

[6] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 272–293, Springer, 2003.

[7] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *International conference on the theory and application of cryptology and information security*, pp. 452–473, Springer, 2003.

[8] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo, "Secure key issuing in id-based cryptography," in *Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation-Volume 32*, pp. 69–74, Citeseer, 2004.

[9] Z. Huang, D. Chen, and Y. Wang, "Multi-signature with anonymous threshold subliminal channel for ad-hoc environments," in *19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers)*, vol. 1, pp. 67–71, IEEE, 2005.

[10] M. L. Das, "A key escrow-free identity-based signature scheme without using secure channel," *Cryptologia*, vol. 35, no. 1, pp. 58–72, 2010.

[11] F. Hess, "Efficient identity based signature schemes based on pairings," in *International workshop on selected areas in cryptography*, pp. 310–324, Springer, 2002.

[12] M. L. Das, A. Saxena, and D. B. Phatak, "Proxy signature scheme with effective revocation using bilinear pairings," *arXiv preprint arXiv:0712.3084*, 2007.