IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 11, 2020, accepted July 3, 2020, date of publication July 15, 2020, date of current version July 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009351

Cryptanalysis and Improvement of a Proxy Signcryption Scheme in the Standard Computational Model

ABDUL WAHEED[1,2], **ARIF IQBAL UMAR**[1], **MAHDI ZAREEI**[3], **(Member, IEEE), NIZAMUD DIN**[4], **NOOR UL AMIN**[1], **JAWAID IQBAL**[1], **YOUSAF SAEED**[5], **AND EHAB MAHMOUD MOHAMED**[6,7], **(Member, IEEE)**

[1]Department of Information Technology, Hazara University Mansehra, Mansehra 21120, Pakistan
[2]School of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea
[3]Tecnologico de Monterrey, School of Engineering and Science, Zapopan 45201, Mexico
[4]Department of Computer Science, University of Chitral, Chitral 17200, Pakistan
[5]Department of Information Technology, University of Haripur, Haripur 25000, Pakistan
[6]Electrical Engineering Department, College of Engineering, Prince Sattam Bin Abdulaziz University, Wadi Addwasir 11991, Saudi Arabia
[7]Electrical Engineering Department, Faculty of Engineering, Aswan University, Aswan 81542, Egypt

Corresponding author: Abdul Waheed (abdul@netlab.snu.ac.kr).

ABSTRACT Proxy signcryption is essential security primitive for emerging secure communication such as e-business, mobile agents, online voting, contract signing, and online auction. It combines the functionality of a proxy signature and encryption to achieve basic security features maintaining a low computational and communicational cost. Ming proposed Proxy Signcryption (PSC) scheme in the standard computational model, claimed it to be secured against: (1) Indistinguishable Chosen Ciphertext Attack (IND-CCA) under the Decisional Bi-linear Diffie-Hellman (DBDH) assumption (2) Existentially Unforgeable Chosen Message Attack (EUF-CMA) under the Computational Diffie Hellman (CDH) assumption. This paper first provides a security analysis to check the correctness and validity of the said PSC scheme. Furthermore, it proves PSC is vulnerable to the launched cryptanalysis attacks. It is established that the PSC is neither semantically secured against IND-CCA nor existentially secured against EUF-CMA in its defined security model. Secondly, we propose an improved new proxy signcryption scheme (N-PSC) based on Elliptic Curve Cryptosystem (ECC) without bi-linear pairing secure against IND-CCA and EUF-CMA for Type-1 adversary $\mathcal{A}_1$ in the standard computational model. It is also proved that the new proposed N-PSC scheme achieves an extra security property of judge verification in case of signature dispute between the proxy correspondents, as well as it outperforms the existing states of the art schemes including the Ming scheme in terms of cost efficiency which makes the new proposed scheme suitable for scarce resources constraint proxy enabled communication applications.

INDEX TERMS Proxy communication, proxy signcryption, cryptanalysis, standard model, elliptic curve cryptography (ECC), IND-CCA, EUF-CMA, third party verification, security model.

I. INTRODUCTION

Privilege delegation, technically known as proxy signcryption mechanism, has become an unavoidable security service in modern enterprises and organizations. It allows a businessperson to extend and operate his business through a designated agent due to temporal absence or lack of time or processing capability. It has applications in e-commerce, such as online proxy auction, mobile agents and business contract signing etc (shown in Fig. 1). Additionally, proxy signcryption (PSC) is a cost-effective mechanism suitable for personal pervasive communication devices like mobile phones, digital assistants having the low computational capability or battery power to perform heavy cryptographic computation where the traditional security techniques are not suitable to satisfy basic security requirements. In 1996, Mambo *et al.* [1], coined the concept of proxy signature that allows an agent called a proxy signer to sign a message on behalf of the

The associate editor coordinating the review of this manuscript and approving it for publication was Shahzad Mumtaz.

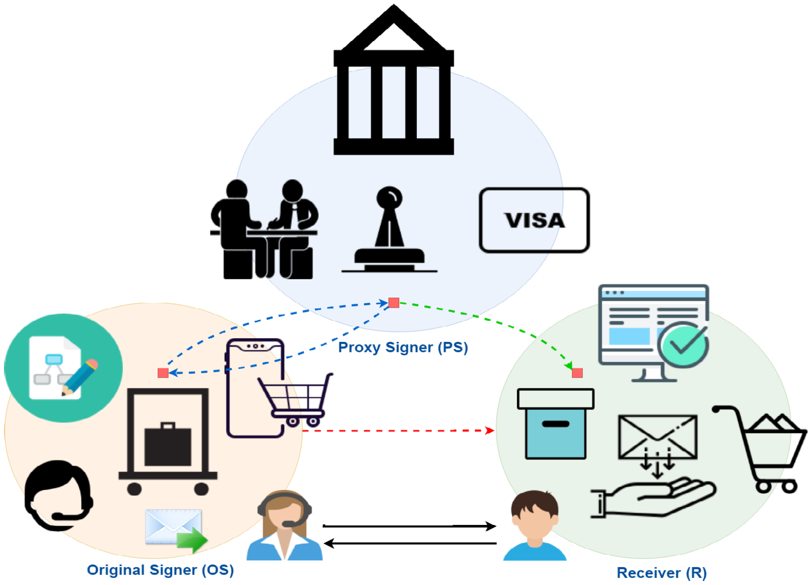


FIGURE 1. Proxy Communication Environment and its Applications.

original signer. Proxy signature has further three categorizations concerning rights delegation, namely, delegation by warrant, partial delegation and full delegation to a proxy signer [2]–[4]. Delegation by the warrant is pivotal and the researchers mostly focused on it in the literature, whereas, the partial and full delegation has several limitations leading to a lack of interest by the researchers' community. Some of the basic security properties like confidentiality, integrity, authenticity and non-repudiation can be achieved using a costly traditional approach, such as, to sign and then encrypt. To overcome the cost issue and reduce the number of machine processing cycles, Zheng [5] proposed the concept of signcryption initially in 1997. It performs the signature and encryption in a single logical step. Following this ground research, several research studies have been focused on signcryption [6]–[15].

In a secure proxy communication, authentication and confidentiality are required to reduce the computation and communication cost with comparison to the individual proxy signature and then encryption concept. This can be achieved by the combined proxy signature and encryption. In 1999, Gamage *et al.* [16] proposed the concept of PSC. In proxy signcryption, the original signer delegates the signing rights to the authorized proxy signer to sign the message on the behalf of him/her. Then, the signed message is returned to the original signer by the proxy signer. Original signer forwards the signcrypted text to legitimate receiver for unsigncryption and verification to confirm whether the message is the original one or not (shown in Fig. 1). For instance, in a company, the boss gives the signing rights to subordinates to sign on behalf of the boss in case of absence. Gamage *et al.* [16], proposed a PSC scheme based on discrete logarithm with no security proofs. In 2003, Jung *et al.* [17] pointed out the limitation of forward secrecy in this scheme [16]. Li and Chen [18], proposed ID based proxy signcryption using bi-linear pairing which was declared insecure by Wang *et al.* [19], having lack of forward secrecy and unforgeability features and presented its improved version as well. Wang and Cao [20] presented two schemes at the same time, first an identity-based proxy signcryption and second a certificateless proxy signcryption. Zhou *et al.* [21] proposed proxy signcryption with the warrant and its security notions using the integer factorization problem. Duan *et al.* [22] presented an ID-based proxy signcryption scheme and its formal security model. Elkamchouchi *et al.* [23] presented the idea of proxy signcryption with signature public verifiability. Lin *et al.* [24] proposed the novel efficient proxy signcryption with practical implementation. Yanfeng *et al.* [25] presented certificateless proxy signcryption (C-PSC) without Bi-linear Pairing (BP) and proved its security and efficiency. In 2017, Bhatia and Verma [26] pointed out the vulnerability of the scheme [25] and presented an improved scheme as well. Ming and Wang [27] proposed a provable proxy signcryption scheme that is vulnerable and compromisable due to its security lapses. Additionally, the flavor of proxy signcryption (like multi-proxy and threshold-proxy) having special function were also proposed and found in literature [28]–[31]. Yu *et al.* [32] recently proposed a new certificateless proxy signcryption scheme using Cyclic Multiplication Groups (CMGs). Li *et al.* [33] proposed a certificateless proxy signcryption scheme for an electronic prescription based on the elliptic curve cryptosystem within the random oracle model.

A. OUR CONTRIBUTION

In literature, several proxy-based signcryption schemes are presented for secure proxy communication in emerging applications. The trends that were followed in this study were focused on different dimensions of novelty and efficiency;

features were added chronologically while comparing a scheme with its preceding schemes for improvement. Ming and Wang [27] claimed that his proposed scheme secure against IND-CCA under the Decisional Bi-linear Diffie Hellman (DBDH) assumption, as well as secure against EUF-CMA under the Computational Diffie Hellman (CDH) assumption. This paper proposes to analyze Ming and Wang proxy signcryption scheme in the standard computational model to check and validate the security vulnerabilities of this scheme by launching the security attack over the said scheme. Moreover, we propose a new improved scheme namely New Proxy Signcryption (N-PSC) based on Elliptic Curve Cryptography (ECC) without bi-linear pairing secured against IND-CCA and EUF-CMA for Type-1 adversary $\mathcal{A}_1$ under the standard computational model and we prove that the new propose N-PSC scheme outperforms than the said scheme as well as a few other existing states of the art schemes in terms of cost efficiency and achieves an extra third party verification (i.e., Judge verification $\mathcal{JV}$) security property in case of any signature dispute which makes the new propose scheme suitable for scarce resources constraint secure proxy communicational environments.

B. PAPER ORGANIZATION

The organization of the rest of this paper is; Section II presents the formal model and some basic definitions with the aspect of proxy signcryption. Section III presents the preliminaries and security notions. Section IV presents a review of Ming's scheme, while, Section V presents the cryptanalysis of Ming's scheme. Section VI proposes an improved N-PSC scheme and Section VII presents its security analysis and finally, the conclusion of the paper is given in Section VIII.

II. SECURE PROXY SIGNCRYPTION

There are three participants in the proxy signcryption scheme, such as Original signer ($\mathcal{OS}$), Proxy signcrypter ($\mathcal{PS}$) and Receiver ($\mathcal{US}$); working together to achieve the security goal in a communication field that shown in Fig. 1:

Original Signcrypter ($\mathcal{OS}$):- The sender of the message has the right to delegate signing rights to proxy signcrypter.

Proxy Signcrypter ($\mathcal{PS}$):- Third party that generates signcrypted text ϑ for $\mathcal{OS}$ on receiving the warrant from $\mathcal{OS}$ and forward to $\mathcal{US}$.

Un-Signcrypter/Receiver ($\mathcal{US}$):- Legitimate receiver of the signcrypted text that verifies and usgncrypt the ϑ.

A. BASIS SECURITY GOALS

Secure proxy signcryption needs to satisfy the following basic security goals [34], [35]:

- *Confidentiality:-* Only the legitimate receivers of the message can unsigncrypt the proxy signcrypted text.
- *Unforgeability:-* It ensures that neither sender nor any third party can create the same valid proxy instead of the authorized one proxy agent.
- *Non-repudiation:-* Receiver can prove easily that received proxy signcrypted text generated by the legitimate third-party/Proxy agent on the behalf of a legitimate/original sender.
- *Forward Secrecy:-* Attacker will not be able to recover the proxy signcrypted text even they know the previous session keys.
- *Verifiability:-* Provides sender and message verification. It convinces receiver on mutual agreement with sender's on signcrypted text, also provides the functionality to verify received message either sent by the legitimate sender or someone else.
- *Proxy key Misuses Prevention:-* After generating proxy signcrypted text, the proxy agent will be unable to use the same session proxy key for other purposes and sessions.

TABLE 1. Notations with description.

Notation	Description
$\mathcal{PSC}$	Proxy Signcryption
$\mathcal{PUSC}$	Proxy Unsigncryption
$\mathcal{OS}$	Proxy Signer
$\mathcal{PS}$	Proxy Signcrypter
$\mathcal{US}$	Proxy Unsigncrypter/Receiver
sk_o, pk_o	Original Signer Private & Public Key
sk_p, pk_p	Proxy Signer Private & Public Key
sk_{us}, pk_{us}	Unsigncrypter Private& Public Key
PKG	Private Key Generator
$\mathcal{M}$	message
$\mathcal{W}$	Warrant Message
$DBDH$	Decision Bi-linear Diffie Hellman
CDC	Computational Diffie Hellman
ECC	Elliptic Curve Cryptography
PKC	Public Key Cryptography
$\mathbb{Z}_P^*$	$\mathbb{G}_1$&$\mathbb{G}_2$ Group with length $p < 2^{160}$
$\perp$	Message authentication failure/Rejects
ϑ	signcrypted text
$\mathbb{G}_1$	cyclic Additive Group
$\mathbb{G}_2$	cyclic Multiplicative Group
$\mathbb{G}$	a Point on Curve
$\mathbb{E}$	Exponentiation
$\mathbb{PM}$	Scalar Point Multiplication
$\mathbb{BP}$	Bi-linear Pairing
$P\&Q$	Two Points on ECC
$\mathcal{C}$	Challenger
$\mathcal{A}$	Attacker
$IND-CCA$	To Check the Confidentiality
$EUF-CMA$	To Check the Forgery

B. NOTATIONS GUIDE

Symbols/notations used throughout this paper are listed in Table 1.

III. PRELIMINARIES

This section describes the concept of bi-linear pairing, several basic hard problems and their security assumptions, types of attackers, proxy signcryption formal communication model and proxy communication basic security notions use against confidentiality and unforgeability as are the following;

Definition 1 (Bi-Linear Pairings ($\mathbb{BP}$) [36]): Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic multiplicative groups with prime order p having generator $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$. Where mapping $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \mapsto \mathbb{G}_T$ is said to be the $\mathbb{BP}$ if hold the properties below:

- *Bi-linearity:- $\hat{e}(g^m, g^n) = \hat{e}(g_1, g_2)^{mn}$ such that m and $n \in \mathbb{G}_1$.*
- *Non-degeneracy:- $\hat{e}(g_1, g_2) \neq 1 \in \mathbb{G}_1$.*
- *Compute-ability:- It is compute-able efficiently $\hat{e}(g_1, g_2)$ such that $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$.*

A. SECURITY HARD PROBLEMS

Proxy signcryption scheme security depends on the hardness of the following problems.

Definition 2 (Computational Diffie-Hellman Problem (CDHP)): Let to assume that $g, g^m, g^n \in \mathbb{G}_1$ and computes g^{mn} such that $m, n \in \mathbb{Z}_p$.

Definition 3 (Computational Diffie-Hellman (CDH) Assumption): We can say that CDHP is (τ, ϱ) hard in $\mathbb{G}_1$ if any τ time algorithm cannot solve with at least ϱ probability.

Adversary $\mathcal{A}$ can solve the CDHP if the probability of chosen random bits are m and n as;

$$Succ_{\mathcal{A}}^{CDH} = Prob[\mathcal{A}\,(g, g^m, g^n){=}g^{mn}] \geq \varrho$$

Definition 4 [Decisional Bi-linear Diffie-Hellman Problem (DBDH-P)]: Let $g, g^m, g^n, g^o \in \mathbb{G}_1$ such that m, n, $o \in \mathbb{Z}_p^*$ and $\mathbb{Z} = (g, g)^{mno}$such that $\mathbb{Z} \in \mathbb{G}_2$.

Definition 5 (Decisional Bi-linear Diffie-Hellman (DBDH) Assumption): A DBDH is (τ, ϱ) hard in $\mathbb{G}_1$ and $\mathbb{G}_2$ if any τ time algorithm cannot solve with at least ϱ probability.

Adversary $\mathcal{A}$ gets advantages polynomial times against DBDH-P as;

Where,

$$|Prob[\mathcal{A}(g, g^m, g^n, g^o, \hat{e}(g, g)^{mno}){=}1] - Prob[\mathcal{A}(g, g^m, g^n, g^o, \mathbb{Z}){=}1]| \geq \varrho$$

B. ADVERSARY TYPES

This section defines three types of adversary $\mathcal{A}$ that are:

a) *Type-1($\mathcal{A}_1$):-* This type of attacker only knows the original signer and proxy signer public keys.
b) *Type-2 ($\mathcal{A}_2$):-* This type of attacker knows both keys (public and private keys) of the proxy signer and only public key of the original signer.
c) *Type-3 ($\mathcal{A}_3$):-* This type of attacker knows both keys (public and private keys) of the original signer and only public key of the proxy signer.

C. PROXY SIGNCRYPTION FORMAL COMMUNICATION MODEL

$\mathcal{PSC}$ formal communication model has the following phases as shown in Fig. 2. This scheme has a total of five algorithms where three known as probabilistic and two deterministic polynomial time algorithms:

Setup:- This is a Probabilistic Polynomial-time (PPT) algorithm, that takes security parameter k as input and returns system parameters *params*.

Key Generation ($\mathcal{KG}$):- It is a PPT algorithm collects *params* as a input and return couple of keys (Private and Public keys) for each entity. Original Signer $\mathcal{OS}$ key pair is (sk_o and pk_o), for Proxy Signcrypter $\mathcal{PS}$ (pk_p and sk_p) and for Unsigncrypter/Receiver $\mathcal{US}$ (pk_{us} and sk_{us}).

Delegation Generation ($\mathcal{DG}$):- It is also a PPT algorithm run by $\mathcal{OS}$ takes input ($params$, sk_o, $\mathcal{W}$) and returns a delegation ϑ_W and forwards output ($\mathcal{W}$, ϑ_W) to $\mathcal{PS}$.

Delegation Verification ($\mathcal{DV}$):- It is Deterministic Polynomial Time (DPT) algorithm. It is run by $\mathcal{PS}$ collects input parameters ($\mathcal{W}$, ϑ_W) and returns delegation ϑ_W or error symbol $\perp$.

Proxy Signcryption ($\mathcal{PSC}$):- Its is a PPT algorithm runs on $\mathcal{PS}$ side with input ($params$, $\mathcal{W}$, ϑ_W, sk_p, pk_{us}, $\mathcal{M}$) and returns the output as proxy signcrypted text ϑ.

Proxy Unsigncryption ($\mathcal{PUSC}$):- It is a DPT algorithm runs on receiver $\mathcal{US}$ side takes input ($params$, $\mathcal{W}$, ϑ, sk_{us}, pk_o, pk_p) and returns output message $\mathcal{M}$ or $\perp$.

D. PROXY SIGNCRYPTION SECURITY NOTIONS

Confidentiality and unforgeability are the two most important security properties of proxy signcryption schemes. Therefore, this section of the paper discusses *Confidentiality (IND-CCA)* and *Unforgeability (EUF-CMA)* captured by game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}$ as under;

1) SEMANTIC SECURITY AGAINST IND-PSC-CCA2

Proxy signcryption confidentiality property captured using IND-CCA game played between $\mathcal{C}$ and $\mathcal{A}$ as;

Setup:- Using this algorithm challenger $\mathcal{C}$ takes security parameters k and returns *params* and forwards to adversary $\mathcal{A}$.

Phase 01:- At this phase, $\mathcal{A}$ performs an adaptive number of following queries where each query depends on a previous query answer.

$\mathcal{DG}$ Queries:- $\mathcal{A}$ makes request to challenger with delegation on warrant $\mathcal{W}$, then $\mathcal{C}$ generates the delegation generation and returns delegation $\mathcal{W}_\vartheta$ to $\mathcal{A}$.

Proxy Signcryption Queries:- $\mathcal{A}$ makes request ϑ, using $\mathcal{W}$ and $\mathcal{M}$ using pk_{us}, $\mathcal{C}$ returns ϑ to $\mathcal{A}$ after running proxy signcryption algorithm.

Proxy Unsigncryption Queries:- $\mathcal{A}$ requests to $\mathcal{C}$ with parameters ($\mathcal{W}$, ϑ, pk_o, pk_p, pk_{us}) for message $\mathcal{M}$ and $\mathcal{C}$ runs proxy unsigncryption algorithm and returns message $\mathcal{M}$ to $\mathcal{A}$.

Challenge:- At the end of the phase one $\mathcal{A}$ makes a decision and selects two messages of equal length such that $\mathcal{M}_0$ and $\mathcal{M}_1$ and makes a challenge. The Challenger $\mathcal{C}$ randomly chooses a bit ψ and generates ciphertext ϑ^* for M_ψ and provides ϑ^* to $\mathcal{A}$.

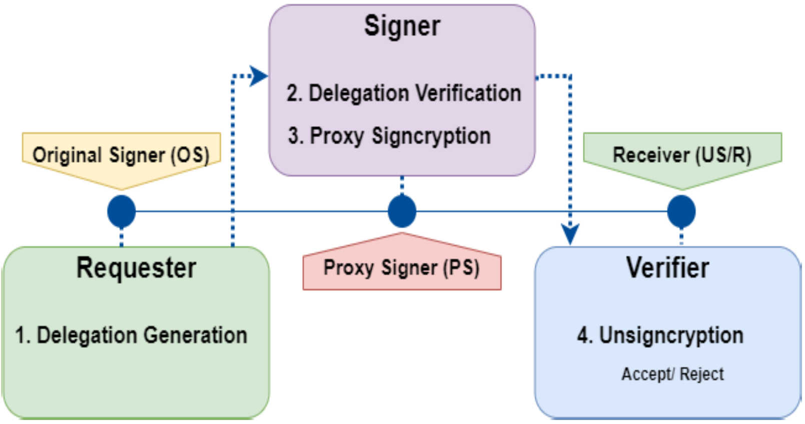


FIGURE 2. Proxy signcryption communication setup.

Phase 02:- Just like phase one, $\mathcal{A}$ starts polynomial time queries without unsigncryption query making on ϑ^* under warrant $\mathcal{W}^*$.

Guess:- At the end of the game, $\mathcal{A}$ generates output bit $\acute{\psi}$ and wins the game where $\acute{\psi} = \psi$.

The $\mathcal{A}$ advantages defined as:

$$\text{Advantage}_{\mathcal{A}}^{IND-PSC-CCA2} = 2\left|\text{Prob}\left[\acute{\psi} = \psi\right] - 1\right|$$

2) EXISTENTIAL UNFORGEABILITY AGAINST EUF-PSC-CMA

To check and prove proxy signcryption scheme existential unforgeability against EUF-PSC-CMA can be viewed by following interactive security game between the challenger $\mathcal{C}$ and forger/adversary $\mathcal{A}$. Access to the desired entities public and private keys depends on the adversary types (i.e., $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$). All the following oracles executed via challenger $\mathcal{C}$.

Setup:- First of all $\mathcal{C}$ runs the setup algorithm then key generation ($\mathcal{KG}$) algorithm to compute system *params*, public and private keys of each individual entity such that $\mathcal{OS}$ (pk_o, sk_o), $\mathcal{PS}$ (pk_p, sk_p) and $\mathcal{US}$ (pk_{us}, sk_{us}) respectively. $\mathcal{C}$ forwards public keys to the adversary $\mathcal{A}_1$.

$\mathcal{PSC}$ Queries:- Using warrant $\mathcal{W}$ attacker $\mathcal{A}_1$ forwards $\mathcal{M}$ and pk_{us} to challenger with a request to send ϑ. Challenger returns the ϑ to attacker under the signcryption algorithm.

$\mathcal{PUSC}$ Queries:- Similar to previous query the attacker $\mathcal{A}_1$ makes request on warrant $\mathcal{W}$ using signcrypted message ϑ for message $\mathcal{M}$ after sending public keys of the entities (pk_o, pk_p, pk_{us}). $\mathcal{C}$ runs $\mathcal{PUSC}$ algorithm to generate unsigncrypt message and then returns it to $\mathcal{A}_1$.

Forgery:- In above interactive game the attacker computes new ϑ^* using warrant $\mathcal{W}^*$ and receiver/unsigncrypter $\mathcal{US}$ public key pk_{us} This defines the winning probability of the attacker $\mathcal{A}_1$ is as;

$$\text{Succ Prob}_{\mathcal{A}_1}^{EUF-PSC-CMA}$$

IV. REVIEW OF MING'S SCHEME

In this section, we review the proxy signryption scheme proposed by Ming in the standard computational model. The scheme consists of the following algorithms;

Setup:- Using setup phase the system generates two multiplicative groups $\mathbb{G}_1$ and $\mathbb{G}_2$ having prime order p with generators g_1 and g_2 of group $\mathbb{G}_1, \mathbb{G}_2$ respectively. System chooses values g_1 and g_2 randomly. Selects security parameter k, defines hash functions that map to $[\mathbb{H}_1, \mathbb{H}_2] : \{0,1\}^* \mapsto [\{0,1\}^n, \{0,1\}^{nv}]$, an admissible bi-linear map $\mathbb{BP}$ such that $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \mapsto \mathbb{G}_T$. Chooses other system's parameters like $v', w' \in \mathbb{G}_1$ such that $v = v_i$ and $w = w_i$, having the length n_v, n_w respectively, then the system publishes these parameters $\{\mathbb{H}_1, \mathbb{H}_2, \mathbb{G}_1, \mathbb{G}_2, g_1, g_2, p, \hat{e}, v, w, v', w'\}$.

Key Generation $\mathcal{KG}$:- In key generation phase each user of PSC scheme chooses random number $sk \in \mathbb{Z}_p$ as a private key and then computes its associated public key in a way such as $pk_{(o,p,us)} = g^{sk_{(o,p,us)}}$. As a result, the public and private key pair of each entity is: (original signer $\mathcal{OS}$ $(pk_o = g^{sk_o}, sk_o)$, proxy signcrypter $\mathcal{PS}$ $(pk_p = g^{sk_p}, sk_p)$, unsigncrypter/receiver $\mathcal{US}$ $(pk_{us} = g^{sk_{us}}, sk_{us})$.

Delegation Generation $\mathcal{DG}$:- For delegation message generation the original signer $\mathcal{OS}$ chooses a random number $r_w \in \mathbb{Z}_p^*$ and generates $\mathcal{W}$ [having length n_w] as a warrant message and sends it to the proxy signcrypter $\mathcal{PS}$ to generate proxy sign on it. $\mathcal{W} \subset \{1, 2, \ldots, n_w\}$ is a set of indices where $[i]$ represents the $i - th$ bits of warrant message and $i = 1 \ni \mathcal{W}[i]$. The following steps are used to compute the warrant delegation message:

1) Chooses randomly $r_w \in \mathbb{Z}_p^*$ and then
2) Computes

$$\vartheta_w = (\vartheta_{w_1}, \vartheta_{w_2}) = (g_1^{sk_o}(w' \Pi_{i \in \mathcal{W}} w_i)^{r_w}, g^{r\mathcal{W}})$$

Sends ϑ_w and $\mathcal{W}$ to the $\mathcal{PS}$.

Delegation Verification $\mathcal{DV}$:- After receiving delegation message (ϑ_w and $\mathcal{W}$) $\mathcal{PS}$ first verifies the received message validity whether the message is the legitimate or not as;

If satisfies $\hat{e}(\vartheta_{w_1}, g) = \hat{e}(g_1, pk_o)\hat{e}(w' \prod_{i \in \mathcal{W}} w_i, \vartheta_{w_2})$ then forwards delegation message to $\mathcal{PS}$ else returns it back to $\mathcal{OS}$ with request to send it again.

Proxy Signcryption $\mathcal{PSC}$:- After finishing the verification phase $\mathcal{PS}$ receives the warrant delegation $\mathcal{W}$ and generates the signature for $\mathcal{M} \in \{0, 1\}^n$ using following steps on the behalf of a legitimate receiver/unsingcrypter ($\mathcal{US}$).

Algorithm 1 Proxy Signcryption $\mathcal{PSC}$

```
1) Randomly chooses $r \in \mathbb{Z}_p$
2) Computes $\vartheta_1 = g^r$
3) Computes $\vartheta_2 = \mathcal{M} \oplus \mathbb{H}_1(\hat{e}(g_1, pk_{us})^r)$
4) Computes $\vartheta_3 = \vartheta_{w_2} = g^{r_w}$
5) Computes $\delta = \mathbb{H}_2(\mathcal{W}, \vartheta_1, \vartheta_2, \vartheta_3, pk_o, pk_p, pk_{us})$
6) Computes $\vartheta_4 = \vartheta_{w_1} g_1^{sk_p} (v' \prod_{i \in \mathcal{V}} v_j)^r$
Forwards output $(\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4)$ to receiver / unsingcrypter ($\mathcal{US}$).
```

Note:- In algorithm above δ [having n_v bits length] and $\mathcal{V}$ [is a set of indices]= subset of $\{1, 2, 3, ..., n_v\}$ where $\delta[j] = 1$ represents $j - bit$ length of δ.

Proxy Unsigncryption $\mathcal{PUSC}$:- Receiver of the message collects message tuples $(\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, pk_o, pk_p, pk_{us}, \mathcal{W} \in \{0, 1\}^{n_w})$ and runs unsigncryption algorithm using the steps following:

Algorithm 2 Proxy Unsigncryption $\mathcal{PUSC}$

```
1) Computes $\mathcal{M} = \vartheta_2 \oplus \mathbb{H}_1(\hat{e}(\vartheta_1, g_1^{sk_{us}}))$
2) Computes $\delta = \mathbb{H}_2(\mathcal{W}, \vartheta_1, \vartheta_2, \vartheta_3, pk_o, pk_p, pk_{us})$
3) The receiver accepts a message if finds the message is
   authentic (i.e., sent by the legitimate
   sender) after satisfying the following equation:
   $$(\vartheta_4, g_1) = e(g_1, pk_o), e(g_1, pk_p)e(w' \prod_{i \in \mathcal{W}} w_i, \vartheta_3)\, e(v' \prod_{i \in \mathcal{V}} v_j, \vartheta_1)$$
```

V. CRYPTANALYSIS OF MING's SCHEME

In this section, we analyze and prove that Ming's scheme cannot resist the cryptanalytic attack and fails to achieve desired security properties for proxy signcryption. Thus, it can be assumed that the adversary can crack the scheme semantic security as well as forges the valid signature. Fig. 3, represents a cryptanalysis attack over a secured communication channel due to which breaks the security mechanism like [37], [38].

A. ATTACK ON SEMANTIC SECURITY IND-PSC-CCA

At this point, we are launching cryptanalytic attack over the Ming's scheme to check its semantic security. We know that adversaries ($\mathcal{A}_1$ and $\mathcal{A}_2$) have capabilities to compute signcryption of the challenged message ($\mathcal{M}_0$ or $\mathcal{M}_1$) with no alteration in message contents which also allows to unsigncrypt it. The adversary yields signcrypted text represented by $\hat{\vartheta} = (\hat{\vartheta}_1, \hat{\vartheta}_2, \hat{\vartheta}_3, \hat{\vartheta}_4)$, which is equal to the challenger signcrypted text $\vartheta^* = (\vartheta_1^*, \vartheta_2^*, \vartheta_3^*, \vartheta_4^*)$. On getting the challenge the $\mathcal{A}_1$ or $\mathcal{A}_2$ generated the signcrypt message $\hat{\vartheta}$ either for $\mathcal{M}_0$ or $\mathcal{M}_1$.

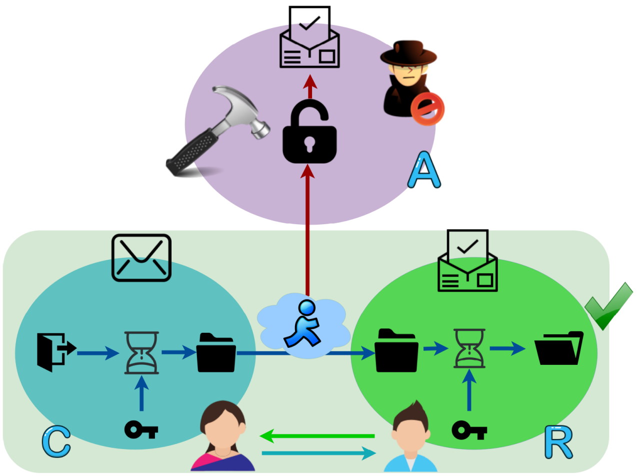


FIGURE 3. Cryptanalysis Attack over a Secured Communication Channel.

Setup:- In this phase the challenger $\mathcal{C}$ runs the machine setup algorithm using input (1^λ) for generation of system parameters $\{\mathbb{H}_1, \mathbb{H}_2, \mathbb{G}_1, \mathbb{G}_2, g_1, g_2, p, \hat{e}, v, w, v', w'\}$ and sends to the adversary $\mathcal{A}_1$.

Phase 01:- $\mathcal{A}_1$ not issue any query to the challenger $\mathcal{C}$.

Challenge:- $\mathcal{A}_1$ chooses two messages of equal size $\mathcal{M}_{0|1}$ and $(pk_o^*, pk_p^*, pk_{us}^*)$ to $\mathcal{C}$ with restriction to ask $\mathbb{O}_{ext}$ query previously. For instance, it gives the same parameters ($\mathcal{M}_{0|1}$ and $pk_o^*, pk_p^*, pk_{us}^*$) to challenger $\mathcal{C}$. Challenger $\mathcal{C}$ accepts the challenge and flips a coin $b \in \{1, 0\}$, after generating signcrypted message ϑ^* on $\mathcal{M}_b^*$ gives to $\mathcal{A}_1$ using the following steps. We need to highlight that the purpose of the challenger is to guess the value of b correctly.

$\mathcal{C}$ first runs the following algorithm steps as,

Algorithm 3 Challenger $\mathcal{C}$ $\mathcal{PSC}$

```
• Randomly chooses $r^* \in \mathbb{Z}_p$
• Computes $\vartheta_1^* = g^{r^*}$
• Computes $\vartheta_2^* = \mathcal{M}_b^* \oplus \mathbb{H}_1(\hat{e}(g_1, pk_{us}^*)^{r^*})$
• Computes $\vartheta_3^* = \vartheta_{w_2}^* = g^{r_w^*}$
• Computes $\delta^* = \mathbb{H}_2(\mathcal{W}, \vartheta_1^*, \vartheta_2^*, \vartheta_3^*, pk_o^*, pk_p^*, pk_{us}^*)$
• Computes $\vartheta_4^* = \vartheta_{w_1}^* g_1^{sk_p^*} (v' \Pi_{i \in \mathcal{V}} v_j)^{r^*}$
Return the signcrypted text $\vartheta^* = (\vartheta_1^*, \vartheta_2^*, \vartheta_3^*, \vartheta_4^*)$
```

Phase 02:- In phase 02, the adversary $\mathcal{A}$ (*Type* 1 *or* 2) first randomly chooses integers $\hat{r} \in \mathbb{Z}_p$ and defines another signcrypted text $\hat{\vartheta} = (\hat{\vartheta}_1, \hat{\vartheta}_2, \hat{\vartheta}_3, \hat{\vartheta}_4)$ and sends to the Challenger.

Algorithm 4 Adversary $\mathcal{A}$ $\mathcal{PSC}$

```
• Computes $\hat{\vartheta}_1 = \vartheta_1^* = g^{\hat{r}}$
• Computes $\hat{\vartheta}_2 = \mathcal{M}_b^* \oplus \mathbb{H}_1(\hat{e}(g_1, pk_{us}^*)^{\hat{r}})$
• Computes $\hat{\vartheta}_3 = \vartheta_3^*$
• Computes $\hat{\delta} = \mathbb{H}_2(\mathcal{W}, \hat{\vartheta}_1, \hat{\vartheta}_2, \hat{\vartheta}_3, pk_o^*, pk_p^*, pk_{us}^*)$
• Computes $\hat{\vartheta}_4 = \hat{\vartheta}_{w_1}(\frac{\vartheta_4}{\vartheta_{w_1}(\vartheta_1)^{(v'+\sum_{i\in\mathcal{V}} v_j)}})(v'\Pi_{i\in\mathcal{V}} v_j)^{\hat{r}}$
  $\therefore$ Adversary $\mathcal{A}$ can compute $g_1^{sk_p}$ using equation 1.
Adversary returns $\hat{\vartheta} = (\hat{\vartheta}_1, \hat{\vartheta}_2, \hat{\vartheta}_3, \hat{\vartheta}_4)$.
```

The $\hat{\vartheta}$ confirms the output is valid and is an identical $\mathcal{PSC}$ if it is compared with challenged message $\mathcal{M}_b^*$. For the said purpose the $\hat{\vartheta}$ above forwards to receiver/unsigncrypter $\mathcal{US}$.

Algorithm 5 $\mathcal{PUSC}$ for Adversary $\mathcal{A}$ Signcryption

```
1) Computes $\hat{\vartheta}_1 = g^{\hat{r}}$
2) Computes $\mathcal{M}_b^* = \hat{\vartheta}_2 \oplus \mathbb{H}_1(\hat{e}(g_1, g_1^{sk_{us}})^{\hat{r}})$
   $\therefore$ Adversary $\mathcal{A}$ calculates $\mathcal{M}_b^*$ as;
   Computes $\hat{\vartheta}_2 = \mathcal{M}_b^* \oplus \mathbb{H}_1(\hat{e}(g_1, g_1^{sk_{us}})^{\hat{r}}) \oplus \mathbb{H}_1(\hat{e}(g_1, g_1^{sk_{us}})^{\hat{r}})$
3) Computes $\delta = \mathbb{H}_2(\mathcal{W}, \hat{\vartheta}_1, \hat{\vartheta}_2, \hat{\vartheta}_3, pk_o, pk_p, pk_{us})$
4) The receiver accepts a message if finds the message is authentic (i.e., sent by the legitimate
   sender) after satisfying the equation following,
   $$(\hat{\vartheta}_4, g_1) = \hat{e}(g_1, pk_o), \hat{e}(g_1, pk_p)\hat{e}(w' \prod_{i\in\mathcal{W}} w_i, \hat{\vartheta}_3)$$
   $$\hat{e}(v' \prod_{i\in\mathcal{V}} v_j, \hat{\vartheta}_1)$$
```

Adversary $\mathcal{A}_1$ makes unsigncryption query from $\mathcal{US}$ and expects that $\hat{\vartheta} \neq \vartheta^*$. The $\mathcal{US}$ responds with unsigncrypted message as $\hat{\vartheta}_2 = \mathcal{M}_b^* \oplus \mathbb{H}_1(\hat{e}(g_1, g_1^{sk_{us}})^{\hat{r}})$ to $\mathcal{A}_1$. After receiving it then $\mathcal{A}_1$ computes the challenged message $\mathcal{M}_b^* \oplus \mathbb{H}_1(\hat{e}(g_1, g_1^{sk_{us}})^{\hat{r}} \oplus \mathbb{H}_1(\hat{e}(g_1, g_1^{sk_{us}})^*)^{\hat{r}})$ and finally finds the message $\hat{\vartheta} = \vartheta^*$ which is also equal to b. It is proved that this is the point for which an attacker tried to win the above confidentiality game and thus we can say the *PSC* scheme is not able to prove the semantic security against indistinguishable security.

B. ATTACK ON UNFORGEABILITY (EUF-PSC-CMA)

Unforgeability means that an unauthorized person cannot change the message signature. In this subsection, we discuss Ming's PSC scheme desired unforgeability property and launch an attack on it to prove whether the scheme is forgeable or not against type 1 adversary $\mathcal{A}_1$ using the following steps.

- **Setup:-** Initially the challenger $\mathcal{C}$ runs machine setup algorithm using input (1^λ) to generate system parameters *params* such that $\{\mathbb{H}_1, \mathbb{H}_2, \mathbb{G}_1, \mathbb{G}_2, g_1, g_2, p, \hat{e}, v, w, v', w'\}$ and sends to the type 1 adversary $\mathcal{A}_1$.
- **Phase 01:-** This phase is just like the above $IND-PSC-CCA$ game.
- **Forgery:-** Using public keys (of $\mathcal{OS}, \mathcal{PS}, \mathcal{US}$) such that pk_o, pk_p, pk_{us} challenger $\mathcal{C}$ desires for signcryption oracle on a message $\mathcal{M}$ where $v' = g^v$ and $v_j = g^{v'_j}$. The adversary $\mathcal{A}_1$, returns with the output $\vartheta = (\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4)$ after computing in the following manner; $g_1^{sk_{(p,o,us)}}$ is the signing parameter computed by adversely $\mathcal{A}_1$ before to sign a message such that,

$$\begin{aligned}
g_1^{sk_{(o,p,us)}} &= \frac{\vartheta_4}{\vartheta_{w_1}(\vartheta_1)^{(v'+\sum_{i\in\mathcal{V}} v_j)}} \\
\frac{\vartheta_4}{\vartheta_{w_1}(\vartheta_1)^{(v'+\sum_{i\in\mathcal{V}} v_j)}} \\
&= \frac{\vartheta_{w_1} g_1^{sk_{(o,p,us)}} (\acute{v}\Pi_{i\in\mathcal{V}} v_{j'})^r}{\vartheta_{w_1}(g_1)^{(v'+\sum_{i\in\mathcal{V}} v_{j'})^r}} \\
&= \frac{g_1^{sk_{(o,p,us)}} (\acute{v}\Pi_{i\in\mathcal{V}} v_{j'})^r}{(g_1)^{(v'+\sum_{i\in\mathcal{V}} v_{j'})^r}} \\
&= \frac{g_1^{sk_{(o,p,us)}} (\acute{v}\Pi_{i\in\mathcal{V}} v_{j'})^r}{(g_1)^{(v)}\Pi_{i\in\mathcal{V}}(g_1^{v_{j'}})^r} \\
&= \frac{g_1^{sk_{(o,p,us)}} (\acute{v}\Pi_{i\in\mathcal{V}} v_j)^r}{(\acute{v}\Pi_{i\in\mathcal{V}} v_j)^r} \\
&= g_1^{sk_{(o,p,us)}} \qquad (1)
\end{aligned}$$

After computing the above signing parameter $(g_1^{sk_{(o,p,us)}})$, $\mathcal{A}_1$ computes the signature as a $\mathcal{PS}$ against the sending message on behalf of original signer $\mathcal{OS}$ and unsigncrypt as a receiver/$\mathcal{US}$ that proves the forgery of Ming's PSC scheme.

VI. PROPOSED SCHEME

This section introduces a new Proxy Signcryption (N-PSC) scheme based on ECC using the standard computational model reflected in Fig. 4. The newly proposed scheme applies to resource-constrained low-computing mobile devices in the standard computational model. A sender/original signer ($\mathcal{OS}$) delegates the responsibility to a proxy signer ($\mathcal{PS}$). The proxy signer communicates to the final destination/$\mathcal{US}$ on behalf of the original signer ($\mathcal{OS}$). The receiver verifies the original signer ($\mathcal{OS}$) through the digital signatures.

The proposed scheme consists of several polynomial-time algorithms outlined below:

A. PROPOSED SCHEME CONSTRUCTION

This section discusses the proposed N-PSC scheme detail that how it works;

Definition 6 [Elliptic-Curve Discrete Logarithmic Problem (ECDLP)]: Let we assume two points on elliptic curve (E_P) *P and Q such that* $Q = k \cdot P$ *where* $k \in \{1, 2, 3, \ldots, n\}$ *is an integer value. Let to assume the value of* $k = 3$ *then* $Q = 3 \cdot P = P + P + P$ *known to be point addition on elliptic curve equal to solve ECDLP [shown in Fig. 5].*

Setup:- Two groups $\mathbb{G}_1$ having generator g_1 and $\mathbb{G}_2$ having the generator g_2 of p prime order. System choosing

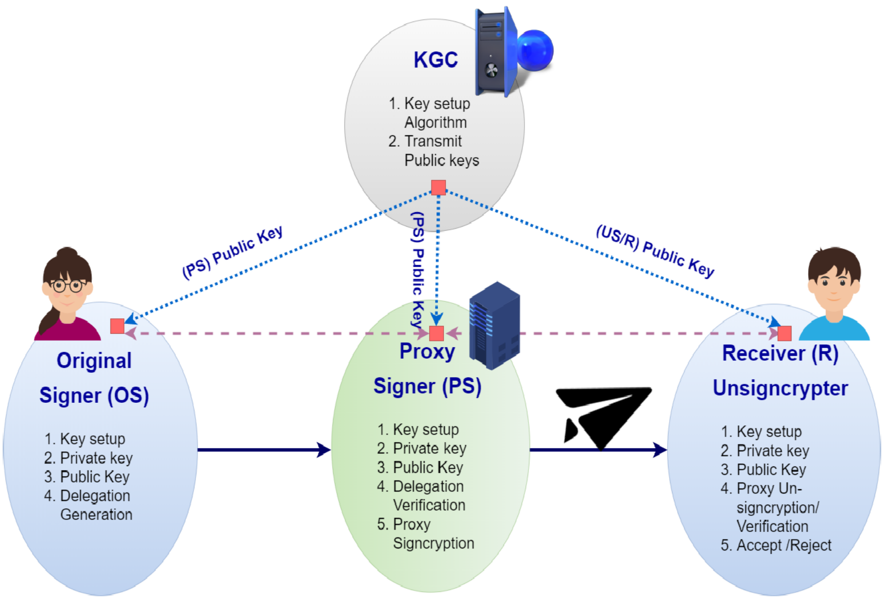


FIGURE 4. New Proposed PSC Scheme.

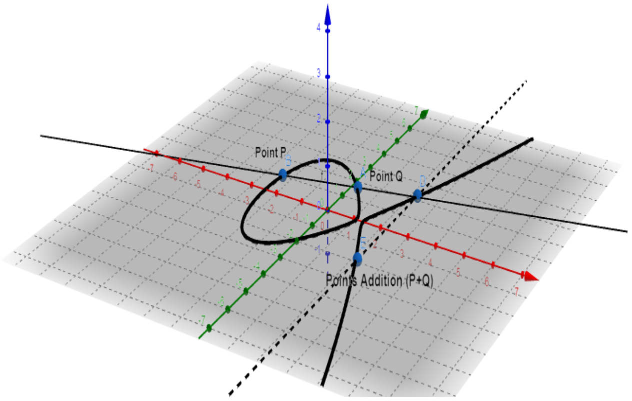


FIGURE 5. Points addition on E_P ECC plane.

g_1 and g_2 randomly and selects k as a system security parameter where hash functions map as $[\mathbb{H}_1, \mathbb{H}_2] : \{0, 1\}^* \mapsto [\{0, 1\}^n, \{0, 1\}^{nv}]$. KGC runs the security parameters with ECC system $\mathcal{E}$ over large prime P under the F_P and $\mathbb{G}$ as a point on curve. Afterwords, the above the system publishes and returns the system's parameters $\{\mathbb{G}, \mathbb{G}_1, \mathbb{G}_2, g_1, g_2, r, E_p, F_p, \mathcal{Z}\}$. Also it chooses the secret key $\varphi \in \mathbb{Z}_p$ and computes the session key (Master key) as $\varphi \cdot pk_{us}$.

Key Generation $\mathcal{KG}$:-

In this phase each user of N-PSC scheme chooses random number $sk \in \mathbb{Z}_p$ as a private key and then computes its associated public key such as $pk_{(o,p,us)} = sk_{(o,p,us)} \cdot \mathbb{G}$. As a result, the public and private key pair of each entity is: (original signer $\mathcal{OS}$ $(pk_o = sk_o \cdot \mathbb{G}, sk_o)$, proxy signer $\mathcal{PS}$ $(pk_p = sk_p \cdot \mathbb{G}, sk_p)$, unsigncrypter $\mathcal{US}/\mathcal{R}$ $(pk_p = sk_{us} \cdot \mathbb{G}, sk_{us})$).

Delegation Generation $\mathcal{DG}$:- This algorithm accepts the original signer $(\mathcal{OS})$ key pair (pk_o, sk_o) and chooses a random number $r_w \in Zp*$ and generates $\mathcal{W}$ [having length n_w] as a warrant message and sends to the proxy signer $\mathcal{PS}$ to generate proxy sign on it. $\mathcal{W} \subset \{1, 2, ..., n_w\}$ is a set of indices where $\mathcal{W}[i]$ represents the $i - th$ bits of warrant message and $i = 1 \ni \mathcal{W}[i]$. The $\mathcal{OS}$ initiates the process and finds the willingness of $\mathcal{PS}$. The following steps are used to compute the warrant delegation message between $\mathcal{OS}$ and $\mathcal{PS}$ after handshaking for the $\mathcal{PS}$ and $\mathcal{US}$ proxy communication.

Delegation Verification $\mathcal{DV}$:- The $\mathcal{PS}$ computes hash r' of received hash value r using $(\vartheta_1, \vartheta_2)$ after decrypting ϑ_w. Compares the hashed values (received and computed) if finds equal $r' = r$ accepts the delegation ϑ_w and computes the proxy signcryption and rejects if $r' \neq r$ with request to $\mathcal{OS}$ to re-send the new ϑ_w.

Proxy key Generation:- The $\mathcal{PS}$ first receives the delegation ϑ_w from $\mathcal{OS}$ and verifies after decryption ϑ_w. If hash values $r' = r$ are equal then accept $\mathcal{OS}$ a legitimate one else reject after computing $\vartheta_w = D_{sk_p}(\vartheta_1, \vartheta_2 || r)$ and $r' = \mathbb{H}_1(\vartheta_1 || \vartheta_2)$.

Original Signer $\mathcal{OS}$

- Selects randomly integers $\Im_{\in \mathbb{R}}\{1, \ldots, n - 1\}$
- Computes $\vartheta_1 = \Im \cdot \mathbb{G}$
- Computes $\vartheta_2 = \mathbb{H}_1(\mathcal{W} \parallel \vartheta_1)$
- $r = \mathbb{H}_1(\vartheta_1 \parallel \vartheta_2)$
- $\vartheta_w = E_{pk_p}(\vartheta_1, \vartheta_2 \parallel r)$

Share ϑ_w with the $\mathcal{PS}$. The $\mathcal{PS}$ first verifies after decryption ϑ_w and if hash values $r' = r$ then accepts $\mathcal{OS}$ a legitimate one else reject.

Proxy Signcryption $\mathcal{PSC}$:- The $\mathcal{PS}$ takes the message $\mathcal{M}$, unsigncrypter $\mathcal{US}$ public key (pk_{us}), its own private key, chooses random numbers $\varphi, \wp \in \mathbb{Z}_P^*$ and computes detail process as follows; computes the session key

$k_{mk} = \mathbb{H}_1(\varphi \cdot pk_{us})$ and then message ($\mathcal{M}$) encryption and generates the signcrypted text $\vartheta = (\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, \vartheta_5)$.

Algorithm 6 Proxy Signcryption $\mathcal{PSC}$

```
1) Selects randomly integers $\varphi_{\in_{\mathbb{R}}}\{1, \dots, n-1\}$
2) Computes $k_{mk} = \mathbb{H}_1(\varphi \cdot pk_{us})$
3) Selects randomly an integer $\wp_{\in_{\mathbb{R}}}\{1, \dots, n-1\}$
4) Generates $s' = (sk_p + \vartheta_2 \cdot \wp)$
5) Computes $\vartheta_3 = E_{pk_{us}}(\mathcal{M}||s')$
6) Computes $\vartheta_4 = (\frac{\varphi}{\vartheta_2+\Im+s'})\ mod\ p$
7) Computes $\vartheta_5 = \wp \cdot \mathbb{G}$ Sends $\vartheta = (\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, \vartheta_5)$
   to receiver.
```

Proxy Unsigncryption $\mathcal{PUSC}$**:-** The $\mathcal{US}$ receives the signcrypted text $\vartheta = (\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, \vartheta_5)$ from $\mathcal{PS}$ and verifies the validity after running unsigncryption $\mathcal{PUSC}$ algorithm using the following steps, if signcrypted text verifies and finds authentic then accept otherwise reject $\perp$.

Algorithm 7 Proxy Unsigncryption $\mathcal{PUSC}$

```
1) Computes $\psi = sk_{us} \cdot \vartheta_4\ modn$
2) Computes $k_{mk} = \mathbb{H}_1(\psi \cdot (pk_p + \vartheta_2 \cdot (\vartheta_5 + \mathbb{G}) + \vartheta_1))$
3) Computes $\mathcal{M}||s' = D_{sk_{us}}(\vartheta_3)$
4) Computes $\vartheta_2' = \mathbb{H}_1(\mathcal{M}||\ \vartheta_1)$
   If $\vartheta_2' = \vartheta_2$ mean $\mathcal{M}$ is original and accept it, else
   reject.
```

B. PROPOSED SCHEME VERIFICATION

1) N-PSC SCHEME CORRECTNESS

The steps following prove equation correctness used in the unsigncryption algorithm at $\mathcal{US}$ side.

$$\psi \cdot (pk_p + \vartheta_2(\vartheta_5 + \mathbb{G}) + \vartheta_1) = \varphi \cdot pk_{us}$$

Proof:

$$\begin{aligned}
&\psi \cdot (pk_p + \vartheta_2(\vartheta_5 + \mathbb{G}) + \vartheta_1)\\
&\ = \psi \cdot (pk_p + \vartheta_2 \cdot \wp \cdot \mathbb{G} + \vartheta_2\mathbb{G} + \vartheta_1)\\
&\ = sk_{us} \cdot \vartheta_4 \cdot (pk_p + \vartheta_2 \cdot \wp \cdot \mathbb{G} + \vartheta_2 \cdot \mathbb{G} + \vartheta_1)\\
&\ = \frac{\varphi}{\vartheta_2 + \Im + s'}(sk_{us} \cdot (sk_p \cdot \mathbb{G}+\vartheta_2 \cdot \wp \cdot \mathbb{G}+\vartheta_2\mathbb{G} + \Im \cdot \mathbb{G}))\\
&\ = \frac{\varphi \cdot pk_{us}}{\vartheta_2 + \Im + s'}(sk_p + \vartheta_2 \cdot \wp + \vartheta_2 + \Im)\\
&\ = \frac{\varphi \cdot pk_{us}}{(\vartheta_2 + \Im + sk_p + \vartheta_2 \cdot \wp)}(sk_p + \vartheta_2 \cdot \wp + \vartheta_2 + \Im)\\
&\ = \varphi \cdot pk_{us}
\end{aligned}$$

The correctness of the proposed N-PSC scheme proved using the above proof. □

Ming's scheme is susceptible to forgery attack in which an adversary $\mathcal{A}$ can compute private keys $g_1^{sk_o,p,us}$ instead of legitimate one $\mathcal{OS}$, $\mathcal{PS}$ and $\mathcal{US}$. The proposed scheme mathematical correctness and verification ensure that the enhanced N-PSC scheme resilient against these adversary attacks.

2) JUDGE VERIFICATION (JV)

The N-PSC scheme also has an extra property to provide the judge verification (third party verification) if a dispute occurs between two parties ($\mathcal{PS}$ and $\mathcal{US}$) the judge can verify and solve the dispute easily after verifying the proxy signature without knowing the message contents. The signature verification operation takes part as using the following computational steps;

Judge receives the published parameters such as $\mathcal{W}$, s', ϑ_5 for the signature issue settlement between the $\mathcal{PS}$ & $\mathcal{US}$ and verifies the signature as;

- Takes the verification parameters $(\mathcal{W}, s', \vartheta_5, pk_p)$
- Verifies $\mathcal{PS}$ public key pk_p with associated digital public key certificate
- Computes $\vartheta_2 = h(\mathcal{W} \parallel \text{relevant information})$
- Computes $y = (s' \cdot \mathbb{G} - \vartheta_2 \cdot \vartheta_5)$
- If $y = pk_p$, shows the sign generated by legitimate proxy signcryption algorithm on $\mathcal{PS}$ side with public key pk_p.

Theorem 7: The following correctness proof ensures that the verification procedure works properly.

$$s' \cdot \mathbb{G} - \vartheta_2 \cdot \vartheta_5 = pk_p$$

Proof:

$$\begin{aligned}
&s' \cdot \mathbb{G} - \vartheta_2 \cdot \vartheta_5\\
&\quad = (sk_p + \vartheta_2 \cdot \wp)\mathbb{G} - \vartheta_2\vartheta_5\\
&\quad = sk_p \cdot \mathbb{G} + \vartheta_2\wp\mathbb{G} - \vartheta_2\vartheta_5\\
&\quad = sk_p \cdot \mathbb{G} + \vartheta_2\wp\mathbb{G} - \vartheta_2\wp\mathbb{G}\\
&\quad = sk_p \cdot \mathbb{G}\\
&\quad = pk_p
\end{aligned}$$

The resultant factor is public key (pk_p) of $\mathcal{PS}$, proved that the signature generated by the legitimate proxy signcrypter $\mathcal{PS}$. □

VII. SECURITY ANALYSIS

This section proves that the N-PSC scheme is secure against IND-NPSC-CCA and EUF-NPSC-CMA under the hardness of ECCDH in the standard computational model. To provide security proofs, we are using the following theorems.

Theorem 8 (Semantic Security IND-NPSC-CCA): The scheme will be IND-NPSC-CCA secured if no adversary with non-negligible advantages wins the games in the PPT interval.

A. GAME 1st

Lemma 9: Here we assume that the proposed N-PSC can be break by an adversary $\mathcal{A}_1$ making the various queries like $(\Gamma k_{psc}, \Gamma k_{Prk}, \Gamma k_u, \Gamma k_{usc}, \Gamma k_{sk}, \Gamma k_{dg}, \tau, \varrho)$. A proxy unsigncrypt query with τ', ϱ' that can solve the ECCDH hard problem having the advantages ϱ' such that:

$$\varrho' = \frac{\varrho}{5\Gamma k_{usc}\{(1, \dots, n) < \mathbb{Z}_P^*\}}$$
$$(\Gamma k_{psc}, \Gamma k_{Prk}, \Gamma k_u, \Gamma k_{usc}, \Gamma k_{sk}, \Gamma k_{dg}, \tau, \varrho) + \mathbb{PM}(5\Gamma_{USC})$$

Here $\mathbb{PM}$ denotes scalar point multiplication.

Proof: Let we suppose that a PPT adversary $\mathcal{A}_1$ having queries tuple such that $\langle \tau, \Gamma k_{H_1}, \Gamma k_{H_2}, \Gamma k_{psc}, \Gamma k_{Prk}, \Gamma k_u, \Gamma k_{usc}, \Gamma k_{sk}, \Gamma k_{dg}, \varrho \rangle$ breaks ECCDH with advantage ϱ lead to compromise the security of the new proposed N-PSC scheme. Furthermore the challenger $\mathcal{C}$ would be no objection to any query of the adversary $\mathcal{A}_1$ and do response accordingly as per the query of the $\mathcal{A}_1$.

Initialization:- The challenger $\mathcal{C}$ runs the setup algorithm and generates the system parameters $\langle \mathcal{E}, F_P, \mathbb{G}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}_1, \mathbb{H}_2, pk_o, pk_{psc}, pk_{us} \rangle$ and selects a random number $\wp \in Z_P^*$ to compute the secret key such that $\vartheta_5 = \wp \cdot \mathbb{G}$ which is hard to break as equal to solve the ECCDH.

Phase 01:- The challenger $\mathcal{C}$ maintains the list of keys and responds to the queries of adversary $\mathcal{A}_1$ accordingly as follows;

Queries List $\mathcal{QL}$**:-** Challenger $\mathcal{C}$ maintains the queries record of the adversary $\mathcal{A}_1$. After the query of $\mathcal{A}_1$, the challenger checks $\mathcal{QL}$ against the query and responds accordingly after the confirmation such that the adversary $\mathcal{A}_1$ queries for public keys and the $\mathcal{C}$ computes and returns the public keys $\mathcal{PS}$ ($pk_p = sk_p \cdot \mathbb{G}$), receiver's $\mathcal{US}$ ($pk_{us} = sk_{us} \cdot \mathbb{G}$).

Delegation Generation Query $\mathcal{DG}q$**:-** Adversary $\mathcal{A}_1$ queries for delegation generation and the $\mathcal{C}$ checks the queries list $\mathcal{QL}$ fist to find the previous correspondence and then runs delegation generation algorithm. On the basis of previous correspondence confirmation $\mathcal{C}$ returns $\mathcal{DG}$ tuple $\langle pk_o, pk_p, pk_{us}, \mathcal{W}, \tau, params \rangle$ to $\mathcal{A}_1$.

Proxy Key Query $\mathcal{PK}q$**:-** Challenger $\mathcal{C}$ maintains the proxy key queries record for adversary $\mathcal{A}_1$ $\mathcal{PK}_\mathcal{L} = \{pk_o, pk_p, pk_{us}, \mathcal{W}, \tau, v\}$. The adversary $\mathcal{A}_1$ query on $\{pk_o, pk_p, \mathcal{W}\}$ and challenger $\mathcal{C}$ searches for the adversary $\mathcal{A}_1$ query tuple such that $\{pk_o, pk_p, \mathcal{W}\}$ if exist returns the receiver proxy key $\mathcal{R}/\mathcal{US}(pk_{us} = sk_{us} \cdot \mathbb{G})$ to the adversary $\mathcal{A}_1$.

Proxy Signcrypt Query $\mathcal{PSC}q$**:-** The adversary $\mathcal{A}_1$ puts the signcrypt query using the tuple $\{pk_o, pk_p, pk_{us}, \mathcal{W}, \tau\}$ with the capabilities to compute signcrypted text for challenged message (either$\mathcal{M}_0$ or $\mathcal{M}_1$) with the no message contents alteration condition. The $\mathcal{C}$ checks the $\mathcal{PK}_\mathcal{L}$ to confirm and then allows the adversary $\mathcal{A}_1$ to run $\mathcal{PSC}$ to signcrypt the message and $\mathcal{USC}$ algorithm to unsigncrypt against the signcrypted query. The adversary $\mathcal{A}_1$ returns signcrypted text such that $\hat{\vartheta} = (\hat{\vartheta}_1, \hat{\vartheta}_2, \hat{\vartheta}_3, \hat{\vartheta}_4, \hat{\vartheta}_5)$, which is equal to the challenger signcrypted text $\vartheta^* = (\vartheta_1^*, \vartheta_2^*, \vartheta_3^*, \vartheta_4^*, \vartheta_5^*)$. For challenge the $\mathcal{A}_1$ generated the signcrypted text $\hat{\vartheta}$ either for $\mathcal{M}_0$ or $\mathcal{M}_1$. The challenger $\mathcal{C}$ runs the machine setup algorithm using input (1^λ) to generate the system parameters $\{\mathbb{H}_1, \mathbb{H}_2, \mathbb{G}_1, \mathbb{G}_2, g_1, \mathbb{Z}_P^*, \mathcal{E}, F_P\}$ and sends to the adversary $\mathcal{A}_1$.

Furthermore $\mathcal{C}$ first verifies, the warrant message ϑ_w and computes the hash value r' to check the authenticity of the warrant ϑ_w. After warrant message originality, the challenger used the $\langle \mathcal{M}_{0|1}, \mathcal{US}, pk_{us} \rangle$ chooses the random number $\varphi, \wp \in \mathbb{Z}_P^*$ to compute the $k_{mk} = \mathbb{H}_1(\varphi \cdot pk_{us})$ the session key. Using the session key the challenger $\mathcal{C}$ encrypts the message $\mathcal{M}_{0|1}$. After that the challenger $\mathcal{C}$ computes $\vartheta_4^* = (\frac{\varphi}{\vartheta_2^* + \Im + s'})$ mod p and $\vartheta_5^* = \wp \cdot \mathbb{G}$. The challenger $\mathcal{C}$ generates ϑ^* which is equal to $(\vartheta_1^*, \vartheta_2^*, \vartheta_3^*, \vartheta_4^*, \vartheta_5^*)$.

Proxy Unsigncrypt Query $\mathcal{PUSC}q$**:-** Challenger $\mathcal{C}$ verifies the submitted signcrypted text of the message, collects message tuples $(\vartheta_1^*, \vartheta_2^*, \vartheta_3^*, \vartheta_4^*, \vartheta_5^*, pk_o, pk_p, pk_{us}, \mathcal{W} \in \{0, 1\}^{n_w})$ and runs unsigncryption algorithm $\mathcal{PUSC}$. Challenger $\mathcal{C}$ computes $\psi = sk_{us} \cdot \vartheta_4^*$ mod n and the session key $k_{mk} = \mathbb{H}_1(\psi \cdot (pk_p + \vartheta_2^* \cdot (\vartheta_5^* + \mathbb{G}) + \vartheta_1^*))$. After getting the session key decrypts the message $\mathcal{M}$ such that $\mathcal{M}||s' = D_{sk_{us}}(\vartheta_3)$ and accepts the computed message if hash values (computed and received) satisfied then returns message $\mathcal{M}_{0|1}$ otherwise reject $\perp$.

Challenge:- Challenger $\mathcal{C}$ chooses two messages of equal size $\mathcal{M}_{0|1}$ with tuple $\langle pk_o^*, pk_p^*, pk_{us}^* \rangle$ with restriction to ask $\mathbb{O}_{ext}$ query previously. Challenger $\mathcal{C}$ generates and shares the ϑ^* to adversary $\mathcal{A}_1$ after running the signcrypted algorithm. For instance also gives the same parameters $\langle \mathcal{M}_{0|1}$ and $pk_o^*, pk_p^*, pk_{us}^* \rangle$ to adversary $\mathcal{A}_1$. $\mathcal{A}_1$ accepts the challenge and flips a coin $b \in \{1, 0\}$ and generates signcrypted message $\hat{\vartheta}$ on the corresponding $\mathcal{M}_b^*$ and compares it after receiving the challenger $\mathcal{C}$ message ϑ^*.

Phase 02:- Obtaining message ϑ^* the adversary $\mathcal{A}_1$ runs queries to the challenger just like phase 01 without asking any query regarding proxy unsigncryption to reveal the session key and unsigncrypt the received message ϑ^* and gets $\mathcal{M}_{0|1}$ trivially at any point of the game in case $\vartheta^* \neq \hat{\vartheta}$.

Output:- Adversary $\mathcal{A}_1$ computes $\lambda' \in \{0, 1\}$ and wins the IND-NPSC-CCA-I game if $\lambda' = \lambda$.

In case the adversary breaks the proposed N-PSC scheme means breaks an existing algorithm simulated by challenger $\mathcal{C}$ equal to solve the ECCDH problem for instance $pk_{us} = sk_{us} \cdot \mathbb{G}$, $\psi = sk_{us} \cdot \vartheta_4$ and $k_{mk} = h(u \cdot (pk_o + \vartheta_2 \cdot (\mathcal{Z} + \mathbb{G}) + \vartheta_1))$ that are hard to solve. The adversary breaks the proposed N-PSC scheme with non-negligible advantage (ϱ') such that;

$$\left| Prob[\lambda' = \lambda] - 1/2 \right| = \varrho'$$

Where,

$$\varrho' = \frac{\varrho}{5\Gamma k_{usc}\{(1, \ldots, n) < \mathbb{Z}_P^*\}}$$

□

Theorem 10 (Unforgeability EUF-NPSC-CMA): The scheme will be EUF-NPSC-CMA secured if no forgery with non-negligible advantages forges and wins the game.

OR

In the standard computational model, if there exists an adversary $\mathcal{A}$ that can break a scheme existential unforgeability, then there is an algorithm with $\mathcal{C}$ which is hard and is equal to solve the ECCDH problem.

Underlying lemma 11 proves the N-PSC scheme EUF-NPSC-CMA secured under ECCDH hard problem that is;

B. GAME 2nd

Lemma 11: Here we assume that the N-PSC scheme secured against forgery under existential unforgeable chosen

TABLE 2. Time Complexity Comparison.

Scheme	KeyGen ($\mathcal{KG}$)	Proxy Signcryption ($\mathcal{PSC}$)	Proxy Unsigncryption ($\mathcal{PUSC}$)	Operations (Total)	Running time (ms)
EAM [23]	$2\mathbb{PM}$	$2\mathbb{E} + 2\mathbb{BP} + 1\mathbb{PM}$	$2\mathbb{E} + 4\mathbb{BP}$	$4\mathbb{E} + 6\mathbb{BP} + 3\mathbb{PM}$	184
LWY [24]	$1\mathbb{PM}$	$1\mathbb{E} + 1\mathbb{BP} + 4\mathbb{PM}$	$3\mathbb{BP} + 4\mathbb{PM}$	$1\mathbb{E} + 4\mathbb{BP} + 9\mathbb{PM}$	148.66
QTLG [25]	$2\mathbb{PM}$	$5\mathbb{PM}$	$11\mathbb{PM}$	$17\mathbb{PM}$	114.84
BA [26]	$2\mathbb{PM}$	$3\mathbb{PM}$	$8\mathbb{PM}$	$9\mathbb{PM}$	82.94
MW [27]	$3\mathbb{E}$	$5\mathbb{E} + 1\mathbb{BP}$	$4\mathbb{BP}$	$8\mathbb{E} + 5\mathbb{BP}$	189.65
HZ [32]	$2\mathbb{E} + 1\mathbb{PM}$	$2\mathbb{E} + 1\mathbb{BP} + 1\mathbb{PM}$	$1\mathbb{E} + 5\mathbb{BP} + 1\mathbb{PM}$	$5\mathbb{E} + 6\mathbb{BP} + 3\mathbb{PM}$	195.2
N-PSC	–	$4\mathbb{PM}$	$5\mathbb{PM}$	$9\mathbb{PM}$	57.42

TABLE 3. Security Properties Comparison.

Scheme	Confidentiality Proofs	Unforgeability Proofs	Third Party Signature Verification	Resistive Against Forgery
EAM [23]	NO	NO	NO	YES
LWY [24]	YES	YES	NO	NO
QTLG [25]	NO	NO	NO	NO
BA [26]	YES	YES	NO	YES
MW [27]	YES	YES	NO	NO
HZ [32]	YES	YES	NO	YES
N-PSC	YES	YES	YES	YES

message attack in the standard computational model if a probabilistic polynomial time adversary $\mathcal{A}_1$ making the various queries (like $\Gamma k_{psc}, \Gamma k_{Prk}, \Gamma k_u, \Gamma k_{usc}, \Gamma k_{sk}, \Gamma k_{dg}, \tau, \varrho$) to proxy signcryption with τ', ϱ' for solving the ECCDH hard problem having the advantages ϱ' such that:

$$\varrho' = \varrho/4\Gamma(k_{dg}+k_{PSC})\Gamma k_{PSC}[\Im, \wp, \varphi \in \{1, \ldots n-1\} < \mathbb{Z}_P^*]$$
$$(\Gamma k_{psc}, \Gamma k_{Prk}, \Gamma k_u, \Gamma k_{usc}, \Gamma k_{sk}, \Gamma k_{dg}, \tau, \varrho) + \mathbb{PM}(4\Gamma_{PSC})$$

Here $\mathbb{PM}$ denotes scalar point multiplication.

Proof: Let we suppose that a PPT adversary $\mathcal{A}_1$ having tuple of queries such that $\langle \tau, \Gamma k_{H_1}, \Gamma k_{H_2}, \Gamma k_{psc}, \Gamma k_{Prk}, \Gamma k_u, \Gamma k_{usc}, \Gamma k_{sk}, \Gamma k_{dg}, \varrho \rangle$ breaks ECCDH with advantage ϱ leads to compromise the security of the new proposed N-PSC scheme. Furthermore the challenger $\mathcal{C}$ would be no objection to any query of the adversary $\mathcal{A}_1$ and do response accordingly as per the query of the $\mathcal{A}_1$.

Initialization:- The challenger $\mathcal{C}$ runs the setup algorithm and generates the system parameters $\langle \mathcal{E}, F_P, \mathbb{G}, \mathbb{G}_1, \mathbb{G}_2, pk_o, pk_{psc}, pk_{r|us} \rangle$ and selects a random numbers $\Im, \varphi, \wp \in \mathbb{Z}_P^*$ to compute ϑ_1, k_{mk} and ϑ_5 such that $\vartheta_1 = \Im \cdot \mathbb{G}$ and $k_{mk} = \varphi \cdot \mathbb{G}$ and $\vartheta_5 = \wp \cdot \mathbb{G}$ respectively that are hard to break as equal to solve the ECCDH.

Phase 01:- As like the theorem 8 the $\mathcal{C}$ maintains the list of keys and responds to the queries of adversary $\mathcal{A}_1$ accordingly as follow and allows the adversary $\mathcal{A}_1$ to make tuple of queries $\langle \tau, k_{psc}, \Gamma k_{Prk}, \Gamma k_u, \Gamma k_{sk}, \Gamma k_{dg}, \varrho \rangle$;

Forgery:- The adversary $\mathcal{A}_1$ puts the signcrypt query using the tuple $\{pk_o, pk_p, pk_{us}, \mathcal{W}, \tau\}$ with the capabilities to compute signcrypted text for challenged message (either $\mathcal{M}_0$ or $\mathcal{M}_1$) with the no message contents alteration condition. Adversary $\mathcal{A}_1$ runs $\mathcal{PSC}$ to signcrypt the message algorithm against the signcrypted query. The adversary $\mathcal{A}_1$ returns signcrypted text such that $\hat{\vartheta} = (\hat{\vartheta}_1, \hat{\vartheta}_2, \hat{\vartheta}_3, \hat{\vartheta}_4, \hat{\vartheta}_5)$. The adversary succeeds if the unsigncrypt query for the $\mathcal{PUSC}$ algorithm doesn't show errors.

In case that the adversary breaks the proposed N-PSC scheme means breaks an existing algorithm simulated by challenger $\mathcal{C}$ equals to solve the ECCDH problem. For instance $s' = (sk_p + \vartheta_2 \cdot \wp)$ and $\vartheta_5 = \wp \cdot \mathbb{G}$,$(\varphi \cdot pk_{us})$ that are hard to solve. The adversary breaks the proposed N-PSC scheme with non-negligible advantage (ϱ') such that;

$$\left|Prob[\lambda' = \lambda] - 1/2\right| = \varrho'$$

Where,

$$\varrho' = \varrho/4\Gamma(k_{dg}+k_{PSC})\Gamma k_{PSC}[\Im, \wp, \varphi \in \{1, \ldots n-1\} < \mathbb{Z}_P^*]$$

□

C. EFFICIENCY

To calculate operational cost we mostly count the number of costly operations used in that scheme. These operations are exponentiation ($\mathbb{E}$), scalar multiplication ($\mathbb{PM}$), bi-linear pairing ($\mathbb{BP}$) and remaining operations consider negligible. The conducted experiment was implemented on the hardware platform of ASUS Z-Book with an Intel ® Core TM $i3 - 6100U$ CPU 2.3GHz and 4 GB memory running on 64-bit Windows 10 operating system. According to Cao *et al.* [39], processing time unit for per $\mathbb{PM}$ is 6.38 ms and for unit $\mathbb{E}$ is counted 11.20 ms and one bi-linear pairing $\mathbb{BP}$ is 20.01 ms. Here we measure the operational cost of the proposed N-PSC scheme and compare it with the already existing schemes found in literature [23]–[27], [32]. The algorithmic complexity comparison is reflected in Table 2.

All the security properties compared with existing schemes found in literature [23]–[27], [32] are reflected in Table 3.

VIII. CONCLUSION

This paper analyzed the Ming's proxy signcryption scheme in the standard computational model. This scheme is reviewed and the cryptanalytic attack was launched on it to check and validate the desired security attributes like confidentiality and existential unforgeability. As a result, it is proved that the scheme is compromisable/vulnerable and neither semantically secured against IND-CCA nor existentially secured against EUF-CMA in their defined security model. A pairing free ECC based improved N-PSC scheme is proposed which was analyzed and found comparatively more efficient and secured than the existing schemes found in the literature with the extra property of third party signature verification. The formal security analysis proved that the improved N-PSC scheme is resilient against IND-CCA and EUF-CMA by Type-1 adversary in the standard computational model.

REFERENCES

[1] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," in *Proc. 3rd ACM Conf. Comput. Commun. Secur. CCS*, 1996, pp. 48–57.

[2] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited," in *Information and Communications Security* (Lecture Notes in Computer Science), vol. 1334, Y. Han, T. Okamoto, and S. Qing, eds. Beijing, China: Springer, 1997; 223–232.

[3] B. Lee, H. Kim, and K. Kim, "Strong proxy signature and its applications," in *Proc. Symp. Cryptogr. Inf. Secur. (SCIS)*, Oiso, Japan, 2001, pp. 603–608.

[4] B. Lee, H. Kim, and K. Kim, "Secure mobile agent using strong nondesignated proxy signature," in *Information Security and Privacy (ACISP)* (Lecture Notes in Computer Science), vol. 2119, V. Varadharajan, Y. Mu, eds. Sydney, NSW, Australia Springer, 2001, pp. 474–486.

[5] Y. Zheng, "Digital signcryption or how to achieve cost (signature and encryption) $\ll$cost (signature) + cost (encryption)," in *Advances in Cryptology (CRYPTO)* (Lecture Notes in Computer Science), vol. 1294, B. S. Kaliski, Jr, Ed. Santa Barbara, CA, USA: Springer-Verlag, 1997, pp. 165–179.

[6] L. JM and W. Mao, "Two birds one stone: Signcryption using RSA," in *Topics in Cryptology (CT-RSA)* (Lecture Notes in Computer Science), vol. 2612, M. Joye, Ed. San Francisco, CA, USA: Springer-Verlag, 2003, pp. 211–225.

[7] J. Malone-Lee. *Identity Based Signcryption*. Accessed: Aug. 23, 2019. [Online]. Available: http://eprint.iacr.org/2002/098.pdf

[8] B. Libert and J. J. Quisquater, "A new identity based signcryption scheme from pairings," in *Proc. IEEE Inf. Theory Workshop*, Mar. 2003, pp. 155–158.

[9] C. SSM, Y. SM, H. LCK, and C. KP, "Efficient forward and provably secure ID based signcryption scheme with public verifiability and public ciphertext authenticity," in *Information Security and Cryptology (ICISC)* (Lecture Notes in Computer Science), vol. 2971, J. I. Lim, D. H. Lee, eds. Seoul South Korea: Springer-Verlag, 2004, pp. 352–369.

[10] X. Boyen, "Multipurpose identity based signcryption: A Swiss army knife for identity based cryptography," in *Advance in Cryptology (CRYPTO)* (Lecture Notes in Computer Science), vol. 2729, D. Boneh, Ed. Santa Barbara, CA, USA: Springer Verlag, 2003, pp. 383–399.

[11] L. Chen and J. Malone-Lee, "Improved identity-based signcryption," in *Public Key Cryptography (PKC)* (Lecture Notes in Computer Science), vol. 3386, S. Vaudenay, Ed. Les Diablerets, Switzerland: Springer, 2005, pp. 362–379.

[12] B. PSLM, B. Libert, N. McCullagh, and Q. JJ, "Efficient and provably-secure identity based signatures and signcryption from bilinear maps," in *Advance in Cryptology (ASIACRYPT)* (Lecture Notes in Computer Science), vol. 3788, B. K. Roy, ed. Chennai, India: Springer, 2005, pp. 515–532.

[13] Y. Yu, B. Yang, Y. Sun, and S.-L. Zhu, "Identity based signcryption scheme without random oracles," *Comput. Standards Interface*, vol. 31, no. 1, pp. 56–62, Jan. 2009.

[14] Z. Jin, Q. Wen, and H. Du, "An improved semantically-secure identity-based signcryption scheme in the standard model," *Comput. Electr. Eng.*, vol. 36, no. 3, pp. 545–552, May 2010.

[15] Z. Liu, Y. Hu, X. Zhang, and H. Ma, "Certificateless signcryption scheme in the standard model," *Inf. Sci.*, vol. 180, no. 3, pp. 452–464, Feb. 2010.

[16] C. Gamage, J. Leiwo, and Y. Zheng, "An efficient scheme for secure message transmission using proxy signcryption," in *Proc. 22nd Australas. Conf. Inf. Secur. Privacy (ACISP)* (Lecture Notes in Computer Science), vol. 2119, V. Varadharajan and Y. Mu eds. Sydney, NSW, Australia: Springer, 1999, pp. 420–431.

[17] J. Hy, L. Dh, L. Ji, And C. Ks, "Signcryption schemes with forward secrecy," in *Proc. Inf. Secur. Appl. (WISA)*, Seoul South Korea, Dec. 2001, pp. 403–475.

[18] X. Li and K. Chen, "Identity based proxy-signcryption scheme from pairings," in *Proc. IEEE Int. Conf. onServices Comput. (SCC)*, Sep. 2004, pp. 494–497.

[19] M. Wang, H. Li, and Z. Liu, "Efficient identity based proxy signcryption schemes with forward security and public verifiability," in *Networking and Mobile Computing (ICCNMC)* (Lecture Notes in Computer Science), vol. 3619, X. Lu and W. Zhao, eds. Zhangjiajie, China: Springer, 2005, pp. 982–991.

[20] Q. Wang and Z. Cao, "Two proxy signcryption schemes from bilinear pairings," in *Cryptology and Network Security (CANS)* (Lecture Notes in Computer Science), vol. 3810, Y. Desmedt, H. Wang, Y. Mu, and Y. Li, eds. Xiamen, China: Springer, 2005, pp. 161–171.

[21] Y. Zhou, Z. Cao, and R. Lu, "Constructing secure warrantbased proxy signcryption schemes," in *Cryptology and Network Security (CANS)* (Lecture Notes in Computer Science), vol. 3810, Y. Desmedt, H. Wang, Y. Mu, and Y. Li, eds. Xiamen, China: Springer, 2005, pp. 172–185.

[22] S. Duan, Z. Cao, and Y. Zhou, "Secure delegation-by-warrant ID-based proxy signcryption scheme," in *Computational Intelligence and Security (CIS)* (Lecture Notes in Computer Science), vol. 3802, Y. Hao, J. Liu, Y. Wang, Y. Cheung, H. Yin, L. Jiao, J. Ma, and Y. Jiao, eds. Xi'an, China: Springer, 2005, pp. 445–450.

[23] D. H. Elkamshoushy, A. K. Aboualsoud, and M. Madkour, "New proxy signcryption scheme with DSA verifier," in *Proc. 23th Nat. Radio Sci. Conf. (NRSC)*, Mar. 2006, pp. 1–8.

[24] H.-Y. Lin, T.-S. Wu, S.-K. Huang, and Y.-S. Yeh, "Efficient proxy signcryption scheme with provable CCA and CMA security," *Comput. Math. with Appl.*, vol. 60, no. 7, pp. 1850–1858, Oct. 2010.

[25] Q. Yanfeng, T. Chunming, L. Yu, X. Maozhi, and G. Baoan, "Certificateless proxy identity-based signcryption scheme without bilinear pairings," *China Commun.*, vol. 10, no. 11, pp. 37–41, Nov. 2013.

[26] T. Bhatia and A. K. Verma, "Cryptanalysis and improvement of certificateless proxy signcryption scheme for e-prescription system in mobile cloud computing," *Ann. Telecommun.*, vol. 72, nos. 9–10, pp. 563–576, Sep. 2017.

[27] Y. Ming and Y. Wang, "Proxy signcryption scheme in the standard model," *Secur. Commun. Netw.*, vol. 8, no. 8, pp. 1431–1446, May 2015.

[28] J. Liu and G. Xiao. *Multi-Proxy Multisigncryption Scheme From Pairings*. Accessed: Aug. 24, 2019. [Online]. Available: http://arxiv.org/abs/cs.CR/0509030

[29] S. Lal and T. Singh. *New ID-Based Multi-Proxy Multisigncryption Scheme From Pairings*. Accessed: Aug. 24, 2019. [Online]. Available: http://arxiv.org/pdf/cs/0701044

[30] Z. Xiaoyan, W. Yan, D. Weifeng, and G. Yan, "An improved ID-based multi-proxy multi-signcryption scheme," in *Proc. 2nd Int. Symp. Electron. Commerce Secur.*, May 2009, pp. 466–469.

[31] Y. Sun, C. Xu, F. Li, and Y. Yu, "Identity based multi-proxy multi-signcryption scheme for electronic commerce," in *Proc. 5th Int. Conf. Inf. Assurance Secur.*, Aug. 2009, pp. 281–284.

[32] H. Yu and Z. Wang, "Construction of certificateless proxy signcryption scheme from CMGs," *IEEE Access*, vol. 7, pp. 141910–141919, Sep. 2019, doi: 10.1109/ACCESS.2019.2943718.

[33] L. Li, S. Zhou, K.-K.-R. Choo, X. Li, and D. He, "An efficient and provably-secure certificateless proxy-signcryption scheme for electronic prescription system," *Secur. Commun. Netw.*, vol. 2018, Aug. 2018, Art. no. 7524102, doi: 10.1155/2018/7524102.

[34] G. Swapna, P. V. S. S. N. Gopal, T. Gowri, and P. V. Reddy, "An efficient ID-based proxy signcryption scheme," *Int. J. Inf. Netw. Secur. (IJINS)*, vol. 1, no. 3, pp. 200–206, Jul. 2012.

[35] S. Pradhan and R. K. Mohapatra, "Proxy blind signature based on ECDLP," *Int. J. Eng. Sci. And Technol.*, vol. 3, no. 3, pp. 1–7, Mar. 2011.

[36] A. Waheed, A. I. Umar, N. Din, N. U. Amin, S. Abdullah, and P. Kumam, "Cryptanalysis of an authentication scheme using an identity based generalized signcryption," *Mathematics*, vol. 7, no. 9, p. 782, Aug. 2019.

[37] C. Zhu, G. Wang, and K. Sun, "Improved cryptanalysis and enhancements of an image encryption scheme using combined 1D chaotic maps," *Entropy*, vol. 20, no. 11, p. 843, Nov. 2018.

[38] M. Li, K. Zhou, H. Ren, and H. Fan, "Cryptanalysis of permutation–diffusion-based lightweight chaotic image encryption scheme using CPA," *Appl. Sci.*, vol. 9, no. 3, p. 494, Jan. 2019.

[39] X. Cao, W. Kou, and X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Inf. Sci.*, vol. 180, no. 15, pp. 2895–2903, 2010.

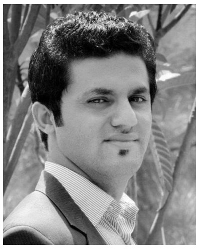
ABDUL WAHEED received the master's degree in computer sciences from the Department of Information Technology, Hazara University Mansehra, in 2014. He is currently pursuing the Ph.D. degree in computer sciences with the Department of Information Technology. He is a member of the Crypto-Net research Group, Hazara University Mansehra. He has completed his Ph.D. research from NetLab-INMC, School of Electrical and Computer Engineering (ECE), Seoul National University (SNU), South Korea, in 2019, under the HEC Research Program. He is also serving as a Lecturer with the Department of Computer Sciences, IQRA National University, Peshawar. He has numerous publications in international conferences and journals. His research interests include information security, secure and smart cryptography, heterogeneous communications within the Internet of Things (IoT), mobile *ad hoc* networks (MANETs), wireless sensor networks (WSNs) security, and fuzzy logic-based decision-making theory.

ARIF IQBAL UMAR received the B.Sc. degree (Hons.) from the Islamia College, Peshawar, Pakistan, in 1991, the M.Sc. degree in computer science from the University of Peshawar, Pakistan, in 1998, and the Ph.D. degree in information retrieval from Beihang University, Beijing, [Beijing University of Aeronautics and Astronautics (BUAA)], China, in 2010. He has at his credit rich experience of more than 27 years of Academic, Research, and Educational Management. He joined Hazara University Mansehra, in 2012. He is leading Head of the Department of Information Technology, Hazara University Mansehra. He has successfully supervised eight Ph.D. Scholars and many more are being trained under his supervision. He has at his credit more than 70 research publications in reputed journals and conferences. He has been a Reviewer of several international journals and conferences. He is a member of several academic bodies of different Universities and has been on organizing bodies of several international conferences. His current research interests include data mining, machine learning, secure and heterogeneous communication in the Internet of Things (IoT), and securing computer networks.

MAHDI ZAREEI (Member, IEEE) received the M.Sc. degree in computer network from the University of Science, Malaysia, in 2011, and the Ph.D. degree from the Communication Systems and Networks Research Group, Malaysia–Japan International Institute of Technology, University of Technology, Malaysia, in 2016. In 2017, he joined the School of Engineering and Sciences, Tecnologico de Monterrey, as a Postdoctoral Fellow, where he has been a Research Professor, since 2019. His research interests include wireless sensor and *ad hoc* networks, energy harvesting sensors, information security, and machine learning. He is a member of the Mexican National Researchers System (level I). He is also serving as an Associate Editor for IEEE Access and *ad hoc*, and Sensor Wireless Networks Journals.

NIZAMUD DIN received the M.Sc. degree from the University of Peshawar, in 2007, the M.S. degree from International Islamic University, Islamabad, in 2012, and the Ph.D. degree in computer Science from Hazara University Mansehra, in 2016. He has published one book and more than 35 research papers in different conferences and journals of international repute. He is also working as an Assistant Professor with the Department of Computer Science, University of Chitral. His current research interests include cryptography, *ad hoc* and sensor network security, electronic voting security, and secure communications in the Internet of Things.

NOOR UL AMIN received the master's degree in computer science from the University of Peshawar, Pakistan, in 1996, and the Ph.D. degree in computer science from the Department of Information Technology, Hazara University Mansehra, Pakistan. He has been the Head of the Department of Information Technology and the Director of IT with Hazara University Mansehra, for 11 years. He is currently the Chair of the Department of Telecommunication, Hazara University Mansehra. He has completed a Research and Development project sponsored by the Ministry of Science and Technology, Pakistan, and established seven hi-tech research and development labs. His research interests include information security, mobile *ad hoc* networks (MANETs), wireless sensor networks (WSNs), and information-centric networking (ICN).



JAWAID IQBAL received the master's degree in computer science from Hazara University Mansehra, Pakistan, in 2014, where he is currently pursuing the Ph.D. degree in computer science with the Department of Information Technology. He is currently a Lecturer with the Department of Information Technology for three years. His research interests include information security, attribute-based signcryption for body sensor networks, and information-centric networking (ICN).

YOUSAF SAEED received the M.S. degree in broadband and high-speed communication networks and the Ph.D. degree in cognitive VANETs from the University of Westminster, London, for which he achieved distinction in individual research thesis on IPv6. He is currently working as an Assistant Professor with the University of Haripur, Pakistan. He received the International Students Award at the College of North West London. His achievements include the publication of monographs, journal articles, and conference papers. His patent is under review regarding emergency vehicles-based traffic lights control systems. He acquired four research projects of the HEC ICT Research and Development 2018.

EHAB MAHMOUD MOHAMED (Member, IEEE) received the B.E. degree in electrical engineering and the M.E. degree in electrical engineering from South Valley University, Egypt, in 2001 and 2006, respectively, and the Ph.D. degree in information science and electrical engineering from Kyushu University, Japan, in 2012. From 2013 to 2016, he was with Osaka University, Japan, as a Specially Appointed Researcher. Since 2017, he has been an Associate Professor with Aswan University, Egypt. He has been an Associate Professor with Prince Sattam Bin Abdulaziz University, Saudi Arabia, since 2019. His current research interests include 5G, B5G and 6G networks, cognitive radio networks, millimeter-wave transmissions, Li-Fi technology, MIMO systems, and underwater communication. He is a technical committee member in many international conferences and a reviewer in many international conferences, journals, and transactions. He is the General Chair of the IEEE ITEMS 2016 and the IEEE ISWC 2018.