mathematics

MDPI

Article

A Novel Provable Secured Signcryption Scheme $\mathcal{PSSS}$: A Hyper-Elliptic Curve-Based Approach

Insaf Ullah [1,2,*], Noor Ul Amin [1], Junaid Khan [3], Muhammad Rehan [3], Muhammad Naeem [3], Hizbullah Khattak [1], Shah Jahan Khattak [4] and Haseen Ali [3]

1 Information Technology Department, Hazra University, Mansehra 21120, Pakistan
2 HIET, Hamdard University Karachi, Islamabad Campuse, Islamabad 44000, Pakistan
3 Departement of Information Technology, Abbotabad University of Science and Technology, Khyber Pakhtunkhwa 22010, Pakistan
4 Pakistan Engineering Council (PEC), Attatruk Avenue (East) Sector G-5/2, Islamabad 44000, Pakistan
* Correspondence: insafktk@gmail.com

Received: 24 March 2019; Accepted: 1 June 2019; Published: 31 July 2019

Abstract: Rivest, Shamir, & Adleman (RSA), bilinear pairing, and elliptic curve are well-known techniques/algorithms for security protocols. These techniques suffer from higher computation and communication costs due to increased sizes of parameters, public keys, and certificates. Hyper-elliptic curve has lower parameter size, public key size, and certificate size. The aim of the proposed work is to reduce the computational cost and communication cost. Furthermore, we validate the security properties of our proposed scheme by using the well-known simulation tool called automated validation of Internet security protocols and applications. Our approach ensures security properties such as resistance against replay attack, confidentiality, authenticity, unforgeability, integrity, non-repudiation, public verifiability, and forward secrecy.

Keywords: signcryption; security properties of signcryption; hyper-elliptic curve; AVISPA

1. Introduction

In the early decades of the Internet, people used particular techniques for secure communication. With its rapid development, security has gradually become more important. Security may be considered in terms of confidentiality, unforgeability, public verifiability, non-repudiation, integrity, authentication, and forward secrecy [1]. Even in the presence of research efforts in the literature, privacy and security measures are still an open challenge [2]. Two elementary cryptographic performances to address the problem of privacy and authenticity in applications were formalized: sign-then-encrypt, encrypt-then-sign, and sign-and-encrypt [3]. Unfortunately, the suggested elementary cryptographic performances require more computational effort. In 1997, Zheng addressed the limitations in elementary performance by introducing a novel technique which he called "signcryption" [3]. Signcryption requires less computational complexity, communication cost, and implementation effort. Moving ahead, Deng and Boa [4] suggested a scheme to reduce computational cost by up to 16% and communication cost by up to 85% when compared to the signature-then-encryption technique. Incongruously, their proposed scheme has more computation and communication costs compared to Zheng's scheme [3]. Gamage et al. [5] improved the approach mentioned in [4] by using firewalls to secure the message without any close-fitting of message confidentiality and deliver clarifications for authentication. This scheme reduces the computation cost to 40% when compared to the customary approach, and its communication cost is similar to Zheng [3]. Despite this, they are incapable of providing forward secrecy. To improve signcryption, several signcryption schemes have been proposed based on the RSA problem [6,7]. Baeket et al. [8] and

then Sharma et al. [9] projected an identity-based signcryption scheme with an extra security service of public verifiability. Furthermore, the new projected scheme in [10] was based on an insubstantial ECC to determine verification and forward secrecy. The scheme needs a limited environment to decrease significant computation and communication costs. Toorani et al. [11] contributed the elliptic curve-based signcryption scheme. They claimed that the proposed signcryption scheme meets a forward secrecy security requirement. Nizamuddin et al. [12] targeted direct public verifiability, while Nizamuddin et al. [13] used one more forward secrecy and public verifiability scheme; however, it essentially needs more assistance with the zero-knowledge protocol. There is work available on the identity-based signcryption schemes in the literature [14–23], but these approaches suffer from the limitations of the key escrow problem. Furthermore, cryptography schemes called certificate-less signcryption have also been contributed [24–33]. However, these schemes need a secure channel for the distribution of partial secret keys, which is still a challenge. Most recently, Elkamchouchi et al. [34] proposed a new signcryption based on elliptic curve cryptography. They claimed to provide all security services in their work. However, unfortunately, their scheme suffers from replay attack, computational cost, and communication cost. Moreover, they did not use any simulation tools for validation purposes. In this paper, we propose a provable secured signcryption scheme based on a hyper-elliptic curve which will provide all security features along with low computational and communication cost. In contrast to the standard cryptosystems such as RSA and bilinear pairing, which use 1024 bits, and elliptic curve, which consumes 160 bits, our scheme uses 80 bits for the parameter size while providing the same level of security along with low computational and communication cost. Furthermore, we validate our scheme by using a well-known simulation tool called automated validation of internet security protocols and applications (AVISPA) in Appendix A [35]. Thus, our approach is appropriate to be used in resource-constrained devices such as sensors, pagers, and smart phones.

1.1. Preliminaries

Assume that g is the genus of the hyper-elliptic curve over a finite field $\mathcal{f}_q$, where the order of this field is q. Furthermore, if $g = 1$, then the group order of $\mathcal{f}_q$ is g. $\log_2 q 2^{160}$. While if $g = 2$, then the curve will require a field $\mathcal{f}_q$ with $|\mathcal{f}_q| \; 2^{80}$, which means that it needs an 80 bit key and parameter size. Suppose $\mathcal{f}^*$ is the algebraic closer of a field $\mathcal{f}$, where $\mathcal{f}$ is a finite field of the hyper-elliptic curve. Therefore, the hyper-elliptic curve of $g > 1$ over $\mathcal{f}$ representing the solution set $(\alpha,\beta) \in \mathcal{f} * \mathcal{f}$ is shown in the following Equation (1):

$$HEC: \; \beta^2 + \mathrm{h}(\alpha)\beta = \; f(\alpha) \bmod q \tag{1}$$

where

- $\mathrm{h}(\alpha) \in \mathcal{f}[\alpha]$ is a polynomial and the degree is $\mathrm{h}(\alpha) \leqslant g$
- $f(\alpha) \in \mathcal{f}[\alpha]$ is the monic polynomial, and the degree is $f(\alpha) \leqslant 2g + 1$
- The points on the hyper-elliptic curve do not form a group unlike an elliptic curve
- The hyper-elliptic curve works on divisor $\mathcal{D}$ which is branded as the formal and finite sum of points on a hyper-elliptic curve that can be further symbolized by Mumford as:

$$\mathcal{D} = (u(\alpha), \; v(\alpha)) = \left(\sum_{i=0}^{g} v_i \alpha^i, \sum_{i=0}^{g-1} v_i \alpha^i \right)$$

Jacobian group $J_c(\mathcal{f}_q)$ is made by divisor from an abelian group and defining the order of Jacobian group $\mathcal{O}(J_c(\mathcal{f}_q))$ as:

$$\left| \left(\sqrt{q} - 1\right)^{2g} \right| \leq \mathcal{O}(\, J_c(\mathcal{f}_q)) \leq \left| \left(\sqrt{q} + 1\right)^{2g} \right|$$

1.2. Hyper-Elliptic Curve Discrete Logarithm (HECDLP)

Suppose there is a divisor $\mathscr{D}$ of order q from the group of Jacobian $(\mathscr{f}_q)$. Also, there is an equation $\mathscr{D}_1 = \mathscr{L}.\mathscr{D}$ *where* $\mathscr{L} \in \mathscr{f}_q$, therefore finding the integer $\mathscr{L}$ is called hyper-elliptic curve discrete logarithm problem.

1.3. Basic Notations

In this section, we discuss the basic notations used in our approach:

$\mathrm{HEC}(\mathscr{f}_q)$:	Represents a hyper-elliptic curve over the field $\mathscr{f}_q$
q:	Is a large prime number and the value of $\|q\| \cong 2^{80}$
$\mathscr{D}$:	Is the divisor of the generalized elliptic curve
$\hbar_1, \hbar_2, \hbar_3$:	Demonstrate the hash functions
$\mathscr{C}$:	Epitomizes the ciphertext
m:	m : Epitomizes the plaintext or message
$M = m.\mathscr{D}$:	Represents the message concatenation with divisor
d_s:	Represents the private key of the signcrypter
$\mathscr{I}_s$:	Represents the public key of the signcrypter
d_u:	Represents the private key of the unsigncrypter
$\mathscr{I}_u$:	Represents the public key of the unsigncrypter
Nr :	Is a fresh nonce
$\mathscr{K}$:	Represent the secret key
$x_{\mathscr{K}}, y_{\mathscr{K}}$:	Representing the subdivided secret key $\mathscr{K}$
E/D:	Represents the encryption and decryption functions
E_{yk} ($E_{\mathrm{xk}}(M)$):	Represents double encryption
D_{yk} ($\mathrm{D}_{\mathrm{xk}}(\mathscr{C})$):	Represents double decryption

2. Formal Model of the Proposed Scheme

In Figure 1, we describe the proposed model of our scheme. Our approach contains four entities—data generator, signcrypter, certificate authority, server and verifier/unsigncrypter. Before starting communication the certificate authority first published the entire public parameters e.g., $\mathscr{D}$, $\hbar_1$, $\hbar_2$, $\hbar_3, \mathscr{E}/\mathscr{D}$, $\mathrm{HEC}(\mathscr{f}_q), \mathscr{f}_q, q$. The data generator verifies the public key of verifier from a certificate authority and then performs the signcryption process with a fresh nonce. After producing the signcryptext, the data generator uploads signcryptext to the server and the server just forwards it to the legitimate receiver/verifier. Later, the verifier first verifies the public key of data generator/signcrypter and then performs the unsigncryption process.

2.1. Proposed Scheme Construction

The proposed $\mathscr{PSSS}$ includes key-generation phase, signcryption algorithm, and unsigncryption algorithm.

2.1.1. Key Generation

The Alice/signcrypter picks a number d_s as a private key from $\{1, \ldots, q-1\}$ and produces their public key $\mathscr{I}_s = d_s.\mathscr{D}$. Also, the Bob or unsigncrypter picks a number d_u from $\{1, \ldots, q-1\}$ and produces their public key $\mathscr{I}_u = d_u.\mathscr{D}$, where $\mathscr{D}$ is the divisor on a hyper-elliptic curve.

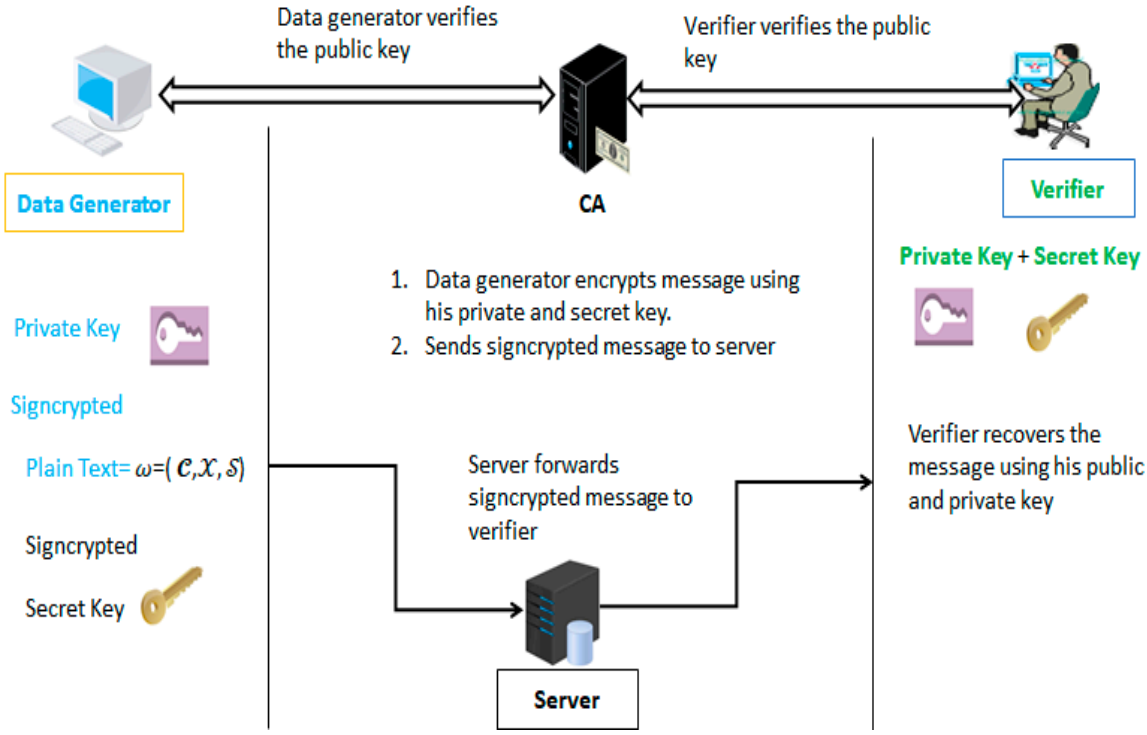

Figure 1. Flow of Proposed Model.

2.1.2. Signcryption

The Algorithm 1, takes the public key of unsigncrypter, divisor, private key of signcrypter and the message $(\mathcal{J}_u, \mathcal{D}, d_s, m)$. After this produces the cipher text and their signature. Sends $\omega = (\mathcal{C}, \mathcal{X}, \mathcal{S})$ to Bob or unsigncrypter.

Algorithm 1. Algorithm $(\mathcal{J}_u, \mathcal{D}, d_s, m)$

Randomly select a number γ from {1, … ...., q − 1}.
Select a nonce Nr
Compute $\mathcal{Z} = \gamma.\mathcal{D} \; mod \; q$ where $\mathcal{D}$ is the divisor on a hyper-elliptic curve.
Divide $\mathcal{Z}$ into x_z, y_z
Compute $\mathcal{X} = \hbar_1(m \,//\, x_z.\mathrm{Nr})$
Compute $\mathcal{K} = \hbar_2((\gamma + \mathcal{X}.d_s).\mathcal{J}_u)$
Divide $\mathcal{K}$ into $x_\mathcal{K}, y_\mathcal{K}$
Compute $\mathcal{M} = m.\mathcal{D}$
Compute $\mathcal{C} = \mathcal{E}y_\mathcal{K}(\mathcal{E}x_\mathcal{K}(\mathcal{M}, \mathrm{Nr}))$
Compute $\mathcal{T} = \hbar_3(\mathcal{C})$
Compute $\mathcal{S} = \gamma^{-1}(d_s\mathcal{X} - \mathcal{T}) mod \; q$
Send ω = $(\mathcal{C}, \mathcal{X}, \mathcal{S})$ to Bob or Unsigncrypter

2.1.3. Unsigncryption

After receiving the signcrypted text ω = $(\mathcal{C}, \mathcal{X}, \mathcal{S})$ this algorithm takes ciphertext, signature, the private key of unsigncrypter, divisor and public key of signcrypter $(\mathcal{C}, \mathcal{X}, \mathcal{S}, d_u, \mathcal{D}, \mathcal{J}_s)$. Therefore, to check the signature authenticity and decrypt the cipher text to plaintext, the unsigncrypter performs unsigncryptions by using the Algorithm 2.

Algorithm 2. Algorithm $(\mathscr{C},\ \mathscr{X}, \mathcal{S}, d_u, \mathscr{D},\ \mathcal{I}_s)$

Compute $\mathcal{T} = \hbar_3(\mathscr{C})$
Compute $\mathcal{Z} = \mathcal{S}^{-1}(\mathscr{X}.\ \mathcal{I}_s - \mathcal{T}.\mathscr{D})$
Divide $\mathcal{Z}$ into x_z, y_z
Divide $\mathcal{Z}$ into x_z, y_z
Use the double decryption method $\mathscr{M}, Nr = \mathscr{D}y_{\mathscr{K}}\left(\mathscr{D}x_{\mathscr{K}}(\mathscr{C})\right)$
Compute $\mathscr{X} = \hbar_1\left(m \,//\, x_z\,.Nr\right)$
Compare $\mathscr{X} = \mathscr{X}$ if equality holds then there is no change in m
Verification of signature is done through $\mathcal{Z}.\ \mathcal{S} + \mathcal{T}.\mathscr{D} = \mathscr{X}.\ \mathcal{I}_s$

3. Correctness

Proof 1. The unsigncrypter can create the secret session key by using the following calculations.

$$\begin{aligned}
&\mathscr{K} = d_u\left(\mathcal{Z} + \mathscr{X}.\mathcal{I}_s\right)\\
&= d_u\left(\mathcal{Z} + \mathscr{X}.\mathcal{I}_s\right) = d_u\left(\gamma.\mathscr{D} + \mathscr{X}.\mathcal{I}_s\right) \textit{ where } \mathcal{Z} = \gamma.\mathscr{D}\\
&= d_u\left(\gamma.\mathscr{D} + \mathscr{X}.d_s.\mathscr{D}\right) \textit{ where } \mathcal{I}_s = d_s.\mathscr{D}\\
&= d_u.\mathscr{D}\left(\gamma + \mathscr{X}.d_s\right) = \mathcal{I}_u\left(\gamma + \mathscr{X}.d_s\right) \textit{ where } \mathcal{I}_u = d_u.\mathscr{D}\\
&= \mathcal{I}_u\left(\gamma + \mathscr{X}.d_s\right) = \mathscr{K}
\end{aligned}$$

□

Proof 2. When any conflict occurs between signcrypter and unsigncrypter, the trusted third party/judge can easily resolve the conflict by performing the following calculation.

$$\begin{aligned}
&\mathcal{Z} = \frac{(\mathscr{X}.\mathcal{I}_s - \mathcal{T}.\mathscr{D})}{\mathcal{S}} = \frac{(\mathscr{X}.\mathcal{I}_s - \mathcal{T}.\mathscr{D})}{\mathcal{S}}\\
&= \frac{(\mathscr{X}.\mathcal{I}_s - \mathcal{T}.\mathscr{D})}{\mathcal{S}} = \frac{(\mathscr{X}.\mathcal{I}_s - \mathcal{T}.\mathscr{D})}{\gamma^{-1}(d_s\mathscr{X} - \mathcal{T})} \text{ where } \mathcal{S} = \gamma^{-1}\left(d_s\mathscr{X} - \mathcal{T}\right)\\
&= \frac{\gamma.(\mathscr{X}.\mathcal{I}_s - \mathcal{T}.\mathscr{D})}{(d_s\mathscr{X} - \mathcal{T})} = \frac{\gamma.(\mathscr{X}.d_s.\mathscr{D} - \mathcal{T}.\mathscr{D})}{(d_s\mathscr{X} - \mathcal{T})} \text{ where } \mathcal{I}_s = d_s.\mathscr{D}\\
&= \frac{\gamma.\mathscr{D}.(\mathscr{X}.d_s - \mathcal{T})}{(\mathscr{X}\,d_s - \mathcal{T})} = \gamma.\mathscr{D} = \mathcal{Z}
\end{aligned}$$

□

4. Security Analysis

In this section, we discuss the security services of our proposed $\mathscr{PSSS}$. Table 1 illustrates the comparisons of the proposed scheme and schemes [10,11,33,34] in terms of security services. Our proposed $\mathscr{PSSS}$ ensures security services such as resistance against replay attack, confidentiality, authenticity, unforgeability, integrity, non-repudiation, public verifiability, and forward secrecy. We validate our scheme by using a well-known simulation tool called AVISPA [35]. AVISPA is an automatic simulation tool for the purpose of validation, verification, and analysis of Internet security-sensitive modules, applications, and cryptographic techniques. With the verification of AVISPA, it becomes stress-free to ensure that the developed protocol is either SAFE or UNSAFE by considering security parameters. To find the results of the designed protocol, it is essential to put them in the form of the High level protocol specification language (HLPSL) language according to its syntax and rules. Code written on the rules of HLPSL language is then converted into lower-level machine language through the intermediate format (IF). The HLPSL2IF translator performs the translation to an intermediate format (IF). According to Dolev and Yao [36,37], HLPSL2IF translator checks the execution in the wisdom of giving initial knowledge, and every agent can construct the messages he is supposed to. AVISPA tool works with four back ends

known as On-the-fly Model-Checker (OFMC), CL-based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC), and Tree-Automata-based Protocol Analyzer (TA4SP) to specify the results. OFMC implements a semi-decision procedure for security of protocols in a bounded number of sessions [38,39]. CL-AtSe represents the set of constraints used to discover if there are attacks on protocols [40]. SATMC [41] is the art of a propositional formula encoding and unrolling of the transition relation to put into a state-of-the-art (SAT) solver to translate back into the attack. The TA4SP [42] illustrates the under-approximation and over-approximation of sessions in protocols by approaching the intruder knowledge with the assistance of regular tree languages. Figure 2 shows the top down flow of AVISPA.

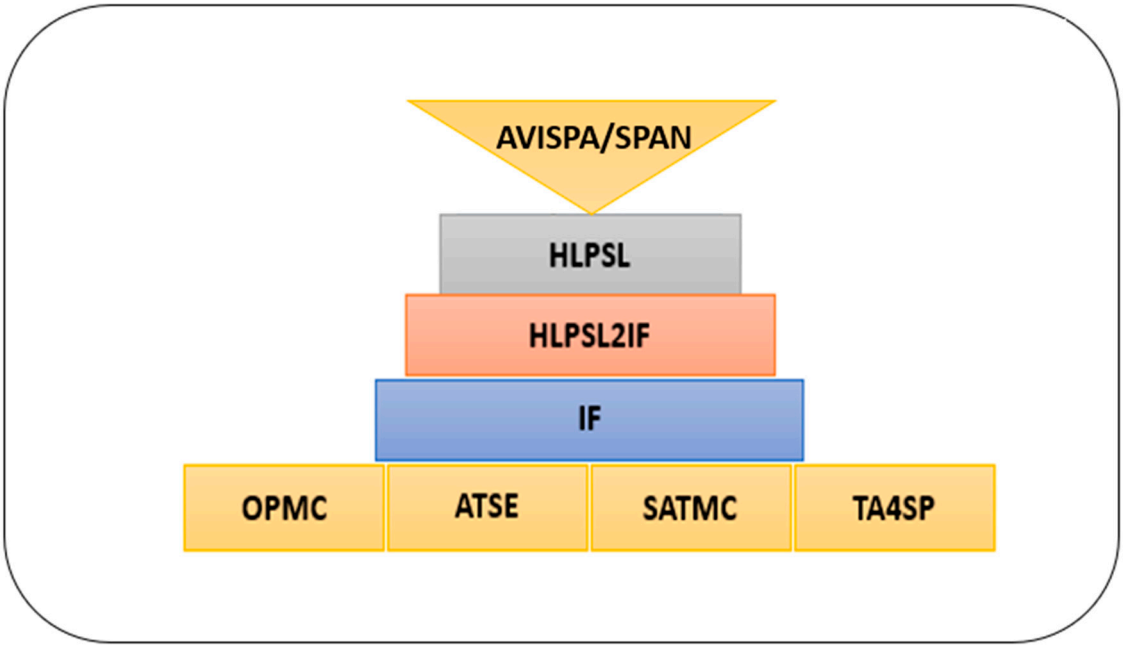

Figure 2. AVISPA Push-bottom Flow.

4.1. Replay Attack

Our scheme ensures that the attacker cannot perform the reply attack based on old messages. If an attacker wants to resend old messages to the unsigncrypter then it generates the tuple $\omega = (\mathscr{C}, \mathscr{X}, \mathscr{S})$ and sends to the unsigncrypter. After receiving a tuple $\omega = (\mathscr{C}, \mathscr{X}, \mathscr{S})$ the unsigncrypter checks the nonce Nr, if it is fresh then the unsigncrypter accepts the tuple otherwise rejects it.

4.2. Confidentiality

Our scheme completely satisfies the security requirement of confidentiality. However, before sending the message to the unsigncrypter, the signcrypter uses the secret key $\mathscr{K}$ to convert the message into ciphertext. Then, an intruder can try to get it from Equation (2). Thus, the intruder must solve Equation (3) and for it must also compute first Equation (4). It is always very challenging to generate the original ciphertext for two hyper-elliptic curve discrete algorithmic problems, so confidentiality of the scheme is maintained.

$$(\mathscr{K} = \mathscr{h}_2\big((\gamma + \mathscr{X}.\mathscr{d}_s).\mathscr{I}_u\big) \tag{2}$$

$$\mathscr{I}_s = \mathscr{d}_s.\mathscr{D} \tag{3}$$

$$\mathscr{Z} = \gamma.\mathscr{D} \tag{4}$$

4.3. Integrity

Our approach calculates the hash value of the message by using the following Equation (5) before delivery. After getting a message the unsigncrypter first checks the freshness of Nr and then computes the hash value from the Equation (6). If freshness checking (Nr) results in $\mathcal{X} = \mathcal{X}$ then integrity of the message is maintained.

$$\mathcal{X} = \hbar_1\left(\hbar \,//\, x_z \,.Nr\right) \tag{5}$$

$$\mathcal{X}^{\mathbb{I}} = \hbar_1\left(\hbar \,//\, x_z \,.Nr\right) \tag{6}$$

4.4. Authenticity

Our scheme ensures the authenticity of signcrypter as the unsigncrypter extract $x_{\mathcal{K}}$, $y_{\mathcal{K}}$ by computing $\mathcal{K} = d_u\left(\mathcal{Z} + \mathcal{X}.\mathcal{I}_s\right) = d_{\mathcal{K}}, y_{\mathcal{K}}$. This is not possible for an intruder to challenge the authenticity of our scheme because they will need the public key of signcrypter and the private key of unsigncrypter to do that. Moreover, unsigncrypter verifies the validity of $x_{\mathcal{K}}$, $y_{\mathcal{K}}$, after decryption of ciphertext (C) and compares computed $x_{\mathcal{K}}$, $y_{\mathcal{K}}$ and decrypted $x_{\mathcal{K}}$, $y_{\mathcal{K}}$ from the cipher text.

4.5. Unforgeability

To produce a forged signature, the forger needs the random private number γ and the private key of signcrypter d_s, as to whom in the following equation. Furthermore, to find out the private random number γ and the private key of a sender d_s, the forger must compute Equations (3) and (4). This is not feasible for a forger to makes two time hyper-elliptic discrete algorithmic problems. Therefore, the unforgeability is maintained in our approach.

$$\mathcal{S}^{\mathbb{I}} = \gamma^{-1}\left(d_s\mathcal{X} - \mathcal{T}\right) \tag{7}$$

4.6. Non-Repudiation

Our scheme assures the security service of non-repudiations. In proposing an approach, the sender generates a digital signature on a message like Equation (7). Hence, in this way, if the signcrypter wants to repudiate from there sent messages the trusted third party can easily identify the source by using the public key of signcrypter.

4.7. Forward Secrecy

Our designed $\mathcal{PSSS}$ meets forward secrecy services of security. In our scheme, even if the private key of a signcrypter is compromised, the attacker cannot decrypt the messages because we use the session secret key for encryption and decryption. Therefore, to generate the secret key $\mathcal{K}$ from Equation (2), the attacker needs the random number γ, which is private to the signcrypter. Thus, to generate $\mathcal{K}$ by using Equation (2) is infeasible and intolerable for attacker to solve the hyper-elliptic curve discrete algorithmic problem. Furthermore, we refresh the secrete key at each session. In this way, our scheme strongly satisfies the forward secrecy property.

4.8. Public Verifiability

Our designed $\mathcal{PSSS}$ ensures public verifiability; to resolve the conflict, a trusted third party can verify the message is from signcrypter or not, by using the following equations.

- Compute $\mathcal{Z} = \mathcal{S}^{-1}(\mathcal{I}_s - \mathcal{T}.\mathcal{D})$
- Compute $\mathcal{X}^{\mathbb{I}} = \hbar_1\left(m \,//\, x_z \,.Nr\right)$
- Compare $\mathcal{X}^{\mathbb{I}} = \mathcal{X}$ if equality holds, then there is no change in m
- Verification of signature is done through $\mathcal{Z}.\mathcal{S} + \mathcal{T}.\mathcal{D} = \mathcal{X}.\mathcal{I}_s$, if satisfy then the message from signcrypter otherwise not.

We have investigated the security services of the designed scheme in terms of a signcryption program. Our designed signcryption scheme accords with the security services for the implementation of both encryption and signatures. We inspected the similarity models of the previous signcryption mechanisms; it is clear that our signcryption scheme offers effective countermeasures in security services, such as schemes [10,33,34], and has been pointed out that the scheme cannot provide resistance against replay attack and not validated through simulation tool; scheme [11] cannot satisfy the services of non-repudiation, and replay attack resistance, and is not validated through simulation tool. By comparison with the previous signcryption mechanisms, the designed method can accomplish all the security services as claimed. Table 1 shows the security requirement comparisons of the proposed scheme and existing schemes [10,11,33,34]. We use the notations $\mathscr{CON}$ for confidentiality, $\mathscr{INT}$ for integrity, $\mathscr{AUT}$ for authentication, $\mathscr{UN}$ for unforgeability, $\mathscr{NRP}$ for non-repudiation, $\mathscr{FRS}$ for forward secrecy, $\mathscr{PV}$ for public verifiability, $\mathscr{RAR}$ for reply attack resistance and $\mathscr{SECVN}$ for security validation by using simulation tools. In addition, we use $\mathscr{YS}$ notation to satisfy the property and $\mathscr{NO}$ not satisfied the property.

Table 1. Comparison in term of security requirements.

Schemes	$\mathscr{CON}$	$\mathscr{INT}$	$\mathscr{AUT}$	$\mathscr{UNF}$	$\mathscr{NRP}$	$\mathscr{FRS}$	$\mathscr{PV}$	$\mathscr{RAR}$	$\mathscr{SECVN}$
Hwang [10]	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{NO}$	$\mathscr{NO}$
Toorani [11]	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{NO}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{NO}$	$\mathscr{NO}$
Mohpathra [33]	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{NO}$	$\mathscr{NO}$
Elkamchouchi [34]	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{NO}$	$\mathscr{NO}$
Proposed	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$	$\mathscr{YS}$

5. Computational Cost

It is important to note that the computational cost is necessary for both sender and receiver. Existing security schemes use elliptic curve point multiplication ($\mathscr{EC}$-$\mathscr{PM}$) which is considered to be a costly operation to measure the computational cost [43]. We use hyper-elliptic curve divisor multiplication ($\mathscr{HC}$-$\mathscr{DM}$) that is far cheaper than ($\mathscr{EC}$-$\mathscr{PM}$) in computational cost. Table 2 shows the computational cost comparisons standard security schemes with the proposed scheme.

Table 2. Comparison of computational cost in terms of major operations.

Schemes	Signcryption	Unsigncryption	Total
Hwang [10]	2 $\mathscr{EC}$-$\mathscr{PM}$	3 $\mathscr{EC}$-$\mathscr{PM}$	5 $\mathscr{EC}$-$\mathscr{PM}$
Toorani [11]	2 $\mathscr{EC}$-$\mathscr{PM}$	3 $\mathscr{EC}$-$\mathscr{PM}$	5 $\mathscr{EC}$-$\mathscr{PM}$
Mohpathra [33]	3 $\mathscr{EC}$-$\mathscr{PM}$	2 $\mathscr{EC}$-$\mathscr{PM}$	5 $\mathscr{EC}$-$\mathscr{PM}$
Elkamchouchi [34]	2 $\mathscr{EC}$-$\mathscr{PM}$	4 $\mathscr{EC}$-$\mathscr{PM}$	6 $\mathscr{EC}$-$\mathscr{PM}$
Proposed	2 $\mathscr{HC}$-$\mathscr{DM}$	4 $\mathscr{HC}$-$\mathscr{DM}$	6 $\mathscr{HC}$-$\mathscr{DM}$

We observed from [44] the experiments were produced hrough by a PC with the following specifications:

- Intel Core i74510UCPU
- 2.0 GHz processor
- 8 GB of memory
- Windows 7 Home Basic
- Multi-precision Integer and Rational Arithmetic C Library (MIRACL)

Their results concluded that single scalar multiplication practices 0.97 ms for elliptic curve point multiplication ($\mathscr{EC}$-$\mathscr{PM}$). We suppose that, if using the same environment like [44], the hyper-elliptic

curve divisors scalar multiplication ($\mathscr{HC}$-$\mathscr{DM}$) will be consumes 0.48 ms because it is the generalized form of elliptic curves and uses the half amount of key,i.e., 80 bits. The following Table 3 shows the computational cost comparisons with respect to time- consuming in milliseconds:

Table 3. Comparative computational cost in term of milli seconds (ms).

Schemes	Signcryption	Unsigncryption	Total
Hwang [10]	1.94	2.91	4.85
Toorani [11]	1.94	2.91	4.85
Mohpathra [33]	2.91	1.91	4.85
Elkamchouchi [34]	2.91	3.88	5.86
Proposed	0.96	1.92	2.88

The computational cost is calculated by using the formula $\frac{\text{existing scheme} - \text{proposed scheme}}{\text{existing scheme}}$ [45]. As shown in Table 3, we observed that the schemes [10,11,33] have the same computational cost which is 4.85 ms, scheme [34] is 5.86 ms and proposed scheme is 2.88 milliseconds.

6. Communication Cost

The minimum communication overhead is the essential requirement of a cryptographic technique for wireless networks. In our approach, we assume that for elliptic curve $|\mathscr{H}||\rho|$ where ρ is a large prime number $\geq 2^{160}$ and for hyper-elliptic curve $|\mathscr{H}|\ |q|$ where q is a large prime number $\geq 2^{80}$. Table 4 elaborates the comparisons in terms of cipher text size and additional parameters. The communication cost of schemes [10,11,33] is $\mathscr{C} + |\mathscr{H}| + 2|\rho|$. Also the communication cost of a scheme [34] is $\mathscr{C} + |\mathscr{H}| + |\rho|$ and proposed scheme is $\mathscr{C} + |\mathscr{H}| + |q|$.

Table 4. Shows the communication cost of proposed and existing signcryption.

Schemes	Communication Cost
Hwang [10]	$\mathscr{C} + \|\mathscr{H}\| + 2\|\rho\|$
Toorani [11]	$\mathscr{C} + \|\mathscr{H}\| + 2\|\rho\|$
Mohpathra [33]	$\mathscr{C} + \|\mathscr{H}\| + 2\|\rho\|$
Elkamchouchi [34]	$\mathscr{C} + \|\mathscr{H}\| + \|\rho\|$
Proposed	$\mathscr{C} + \|\mathscr{H}\| + \|q\|$

Generalized Formulas for the Reduction of Communication Cost

Reduction formulas for the communication cost of the proposed scheme in comparison to the schemes of [10,11,33] are computed with the help of the following equation.

$$\frac{\mathscr{C} + |\mathscr{H}| + 2|\rho| - \mathscr{C} + |\mathscr{H}| + |q|}{\mathscr{C} + |\mathscr{H}| + 2|\rho|}$$

Also, the reduction formula for the communication cost of the proposed scheme as compared to a scheme [34] is followed by Equation (3).

$$\frac{\mathscr{C} + |\mathscr{H}| + |\rho| - \mathscr{C} + |\mathscr{H}| + |q|}{\mathscr{C} + |\mathscr{H}| + |\rho|}$$

Reduction in communication cost depends upon the selection of parameters and the quantity of data. Table 5 shows the comparisons of communication cost of different sizes of ciphertext and parameters of elliptic curves and hyper-elliptic curve of the proposed and existing

schemes [10,11,33,34]. It is clear from our analysis that our scheme uses 21.27% to 52.63% less communication cost than [10,11,33], and 11.90% to 35.71% less than the technique presented in [34].

Table 5. Communication cost comparisons in terms of size of cipher text.

Schemes	Message Size	128 bits	256 bits	1024 bits
Hwang [10]		128 + |160| + 2|160| = 608	256 + |160| + 2|160| = 736	1024 + |160| + 2|160| = 1504
Toorani [11]		128 + |160| + 2|160| = 608	256 + |160| + 2|160| = 736	1024 + |160| + 2|160| = 1504
Mohpathra [33]		128 + |160| + 2|160| = 608	256 + |160| + 2|160| = 736	1024 + |160| + 2|160| = 1504
Elkamchouchi [34]		128 + |160| + |160| = 448	256 + |160| + |160| = 576	1024 + |160| + |160| = 1344
Proposed		128 + |80| + |80| = 288	256 + |80| + |80| = 416	1024 + |80| + |80| = 1184

7. Applications

In this phase, we discuss the applications of our design approach in e-payment systems during online shopping [46,47]. Online shopping system includes three main roles—customer, bank, and merchant. Figure 3 shows the deployment of our scheme in e-payment during online shopping. In this system, the customer first selects the item to buy according to his choice. After this, the merchant will send a validated payment order to his customer. Furthermore, the customer generates a signcryptext of validated payment and then sends it to the bank. After receiving the signcryptext the bank approves payment validations by checking their account details. The bank encrypts these charges through secret key, stores it in the temporary memory and sends it back to the customer with following instructions

- Confirm payment order validity
- If not valid then EXIT

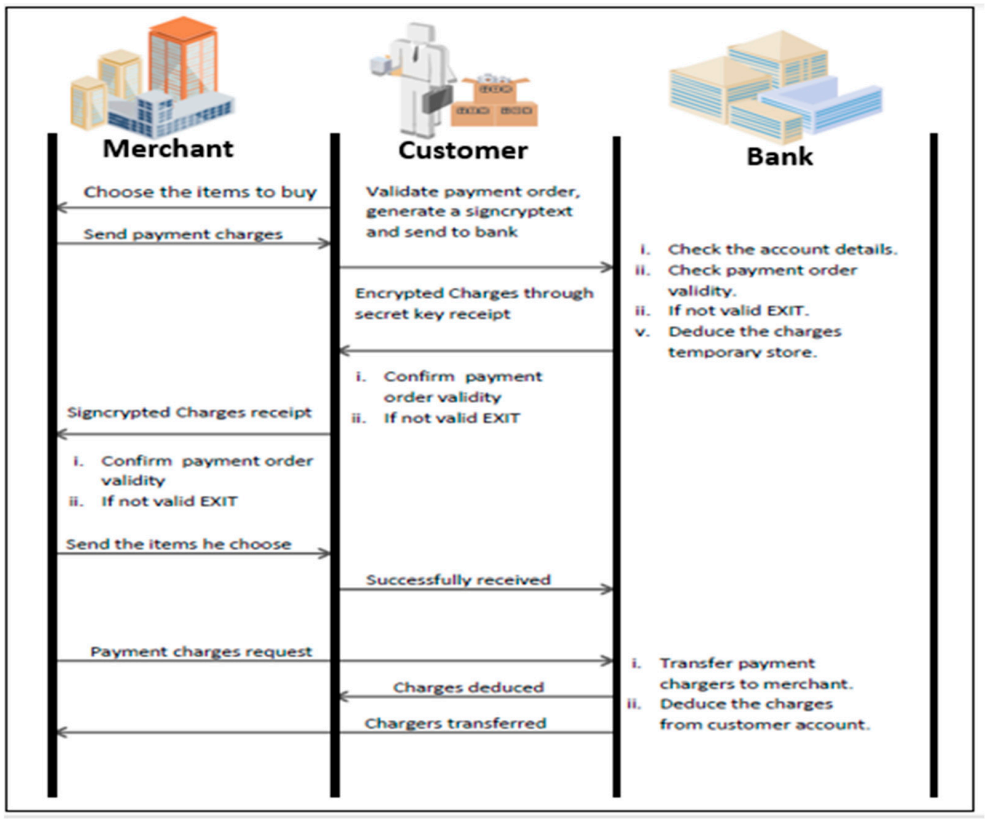

Figure 3. Explanation of application.

For the further process of order, the customer signcrypts these details and sends to the merchant. After confirming the payment order validity, the merchant delivers the particular goods to the customer. After receiving the goods, the customer forwards this acknowledgment to the bank to deduct the charges from his account. To end the whole communication securely, the bank sends payment deduction to the sender as well as to the merchant.

8. Conclusions

This paper presents a new provable secured signcryption scheme ($\mathscr{PSSS}$) based on hyper-elliptic curve. The proposed scheme, reduced in communication cost about 21.27% to 52.63% from [10,11,33], and from [34] is about 11.90% to 35.71%, and reduced in computational cost about 40.61% from [10,11,33], and about 50.85% from [34]. Furthermore, the proposed scheme ensures security properties such as resistance against replay attack confidentiality, authenticity, unforgeability, integrity, non-repudiation, public verifiability, and forward secrecy. Furthermore, we have validated the security properties of our proposed scheme by using well-known simulation tools such as AVISPA.

Author Contributions: Conceptualization, I.U. and N.U.A.; methodology, I.U.; software, H.K., J.K., and M.R.; validation, I.U., J.K., M.R., N.U.A., and M.N.; formal analysis, I.U.; investigation, S.J.K.; writing—original draft preparation, I.U.; writing—review and editing, H.A.; supervision, N.U.A.; project administration, N.U.A.

Funding: This study received no external funding.

Acknowledgments: The authors thanks to anonymous reviewers.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

This section we validate our scheme by using a well-known simulation tool called AVISPA [35]. We generate HLPSL code for our proposed scheme which is shown in Table A1 and tests the result over two protocols OFMC and CL-based Attack Searcher (ATSE). Figures A1 and A2 represent the simulation results of the proposed scheme over OFMC and ATSE, respectively. We cannot use the same notations of our algorithm in HLPSL code due to the classified syntax of the AVISPA tool. The notations used afore to arrow are termed in algorithm and notations procedure after the arrow are used in HLPSL code.

- $\hbar_1$, $\hbar_2$, $\hbar_3$ → H, HH, HHH: hash functions
- m → M: Is plain text or message
- $\mathscr{C}$ → M′: Is the cipher text
- $\mathscr{I}_u$ → Ju: public key of unsigncrypter
- $\mathscr{d}_s$ → inv (Js): private key of signcrypter
- $\mathscr{d}_u$ → inv (Ju): private key of unsigncrypter
- $\mathscr{I}_s$ → Js: public key of signcrypter
- $\mathscr{E}$ → (Signcrypter.{H1(M′.Xz′.Nr)}_inv(Js).{H2(Alpa′.H1(M′.Xz′.Nr).inv(Js)).Ju}_Eyk′. {M′.Nr.Exk′}_Eyk′): encryption
- $\mathscr{D}$ → (Signcrypter.{H1(M′.Xz′.Nr)}_inv(Js).{H2(Alpa′.H1(M′.Xz′.Nr).inv(Js)).Ju}_Eyk′. {M′.Nr.Exk′}_Eyk′): decryption.
- $\mathscr{E}y_{\mathscr{K}}$, $\mathscr{E}x_{\mathscr{K}}$ → Eyk, Exk: Symmetric key
- γ^{-1} → Alpa′: random number
- x_z, y_z → Xz,Yz: Dividing $\mathscr{Z}$
- Ki → intruder public key
- inv (ki) → intruder private key
- $\mathscr{C}$ → {M′.Nr.Exk′}_Eyk′)
- $\mathscr{X}$ → {H1(M′.Xz′.Nr)}_inv(Js)

- $\mathcal{S} \rightarrow$ {H2(Alpa′.H1(M′.Xz′.Nr).inv(Js)).Ju}_Eyk′

In algorithm $\mathcal{S} = \gamma^{-1}\left(\mathcal{d}_s\mathcal{X} - \mathcal{T} \right)$ represents the signature, $\mathcal{E}\mathcal{y}_{\mathcal{K}}\left(\mathcal{E}\mathcal{x}_{\mathcal{K}}(\mathcal{M})\right)$ represents double encryption and $\mathcal{D}\mathcal{y}_{\mathcal{K}}(\mathcal{D}\mathcal{x}_{\mathcal{K}}(\mathcal{C})$ represents double decryption which cannot be signified directly in the code. In HLPSL code Ns: represents nonce for signcrypter, Nr: nonce for unsigncrypter, K: symmetric key, Ki: intruder public key, inv (ki): intruder private key, ki: symmetric key of an intruder.

Table A1. HLPSL CODE.

HLPSL Code ()
role %%start of the protocol by Signcrypter already knows the Unsigncrypter's role_Signcrypter(Signcrypter:agent,Unsigncrypter:agent,Ju:public_key,Js:public_key,Nr:tex t,Ns:text,SND,RCV:channel(dy)) played_by Signcrypter def= local %%start of the protocol by Signcrypter already knowing the %%Unsigncrypter's public key State:nat,Alpa:text,H1:hash_func,Xz:text,H2:hash_func,Eyk:symmetric_key,M:text,Exk :symmetric_key init State:= 0 transition %% signcrypter receives challenge from Unsigncrypter by using his public %%key and nonce, sends message to unsigncrypter in response 1. State=0/\ RCV(Unsigncrypter.{Ns}_Js) =|> State′:=1/\ Eyk′:=new()/\ Exk′:=new()/\ M′:=new()/\ Xz′:=new()/\ Alpa′:=new()/\ SND(Sign crypter.{H1(M′.Xz′.Nr)}_inv(Js).{H2(Alpa′.H1(M′.Xz′.Nr).inv(Js)).Ju}_Eyk′.{M′.Nr.Exk′}_Eyk′)/\ SND(Signcrypter.{H2(Alpa′.H1(M′.Xz′.Nr).inv(Js)).Ju}_Eyk′)/\ SND(Signcrypter.{M′.Nr. Exk′}_Eyk′) end role role %%defining the role played by signcrypter by using its keys … role_Unsigncrypter(Signcrypter:agent,Unsigncrypter:agent,Ju:public_key,Js:public_key,Nr :text,Ns:text,SND,RCV:channel(dy)) played_by Unsigncrypter def= local State:nat,Alpa:text,H1:hash_func,Xz:text,H2:hash_func,Eyk:symmetric_key,M:text,Exk :symmetric_key init State:= 0 transition 1. State=0/\ RCV(start) =|> State′:=1/\ SND(Unsigncrypter.{Ns}_Js) 2. State=1/\ RCV(Signcrypter.{H1(M′.Xz′.Nr)}_inv(Js).{H2(Alpa′.H1(M′.Xz′.Nr).inv(Js)).Ju}_Ey k′.{M′.Nr.Exk′}_Eyk′) =|> State′:=2 3. State=2/\ RCV(Signcrypter.{H2(Alpa.H1(M.Xz.Nr).inv(Js)).Ju}_Eyk) =|> State′:=3 4. State=3/\ RCV(Signcrypter.{M.Nr.Exk}_Eyk) =|> State′:=4 end role role %%session1 between agents signcrypter and unsigncrypter session1(Signcrypter:agent,Unsigncrypter:agent,Ju:public_key,Js:public_key,Nr:text,Ns:text) def=

Table A1. *Cont.*

HLPSL Code ()

```
      local
            SND2,RCV2,SND1,RCV1:channel(dy)
      composition

      role_Unsigncrypter(Signcrypter,Unsigncrypter,Ju,Js,Nr,Ns,SND2,RCV2)/\ role_Signcr
ypter(Signcrypter,Unsigncrypter,Ju,Js,Nr,Ns,SND1,RCV1)
end role
role
%%session2 between agents signcrypter and unsigncrypter
session2(Signcrypter:agent,Unsigncrypter:agent,Ju:public_key,Js:public_key,Nr:text,Ns:text
)
def=
      local
            SND1,RCV1:channel(dy)
      composition
            role_Signcrypter(Signcrypter,Unsigncrypter,Ju,Js,Nr,Ns,SND1,RCV1)
end role

role environment()
def=
      const

      hash_0:hash_func,nr:text,ju:public_key,alice:agent,bob:agent,js:public_key,ns:text,cons
_1: agent,const_2:public_key,const_3:public_key,const_4:text,const_5:text,auth_1:protocol_i
d, sec_2:protocol_id
      intruder_knowledge = {alice,bob}
      composition

      session2(i,const_1,const_2,const_3,const_4,const_5)/\ session1(alice,bob,ju,js,nr,ns)
end role

goal
%% defining the goals as an aim of protocol
      authentication_on auth_1
      secrecy_of sec_2
end goal

environment()
```

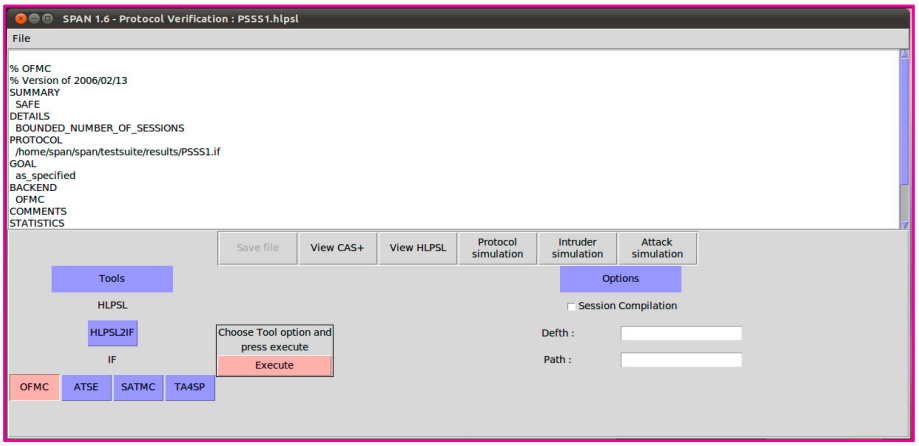

Figure A1. OFMC simulation results.

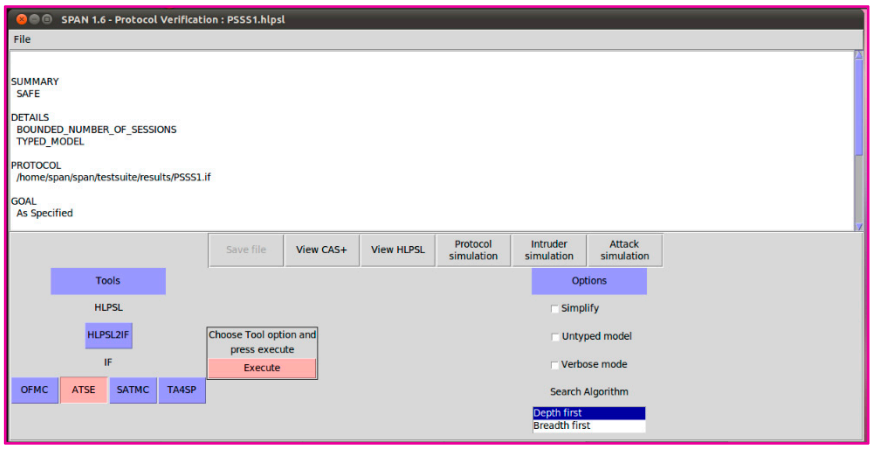

Figure A2. ATSE simulation results.

References

1. Mehmood, A.; Ullah, I.; Noor-Ul-Amin; Umar, A.I. Public Verifiable Generalized Authenticated Encryption (ℙ𝕍ℂÆ) based on Hyper Elliptic Curve. *J. Appl. Environ. Biol. Sci.* **2017**, *7*, 194–200.
2. Sadat, A.; Ahmad, R.; Ullah, I.; Khattak, H.; Ullah, S. Multi Receiver Signcryption Based On Hyper Elliptic Curve Cryptosystem. *J. Appl. Environ. Biol. Sci.* **2017**, *7*, 194–200.
3. Zheng, Y. Digital signcryption or how to achieve cost (signature & encryption) ≪ cost (signature) + cost(encryption). *Lect. Notes Comput. Sci.* **1997**, 165–179. [CrossRef]
4. Bao, F.; Deng, R.H. A signcryption scheme with signature directly verifiable by public key. *Lect. Notes Comput. Sci.* **1998**, 55–59. [CrossRef]
5. Zheng, Y. Identification, signature and signcryption using high order residues modulo an RSA composite. In *Public Key Cryptography*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 48–63.
6. Malone-Lee, J.; Mao, W. Two Birds One Stone: Signcryption Using RSA. *Lect. Notes Comput. Sci.* **2003**, 211–226. [CrossRef]
7. Baek, J.; Steinfeld, R.; Zheng, Y. Formal proofs for the security of signcryption. In *Public Key Cryptography*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 80–98.
8. Sharma, G.; Bala, S.; Verma, A.K. An identity-based ring signcryption scheme. In *IT Convergence and Security*; Springer: Dordrecht, The Netherlands, 2013; pp. 151–157. [CrossRef]
9. Zheng, Y.; Imai, H. How to construct efficient signcryption schemes on elliptic curves. *Inf. Process. Lett.* **1998**, *68*, 227–233. [CrossRef]
10. Hwang, R.-J.; Lai, C.-H.; Su, F.-F. An efficient signcryption scheme with forward secrecy based on elliptic curve. *Appl. Math. Comput.* **2005**, *167*, 870–881. [CrossRef]
11. Toorani, M.; Beheshti, A.A. An elliptic curve-based signcryption scheme with forward secrecy. *arXiv* **2010**, arXiv:1005.1856. [CrossRef]
12. Nizamuddin; Chudry, S.A.; Amin, N. Signcryption schemes withforward secrecybased on hyperelliptic curve cryptosystem. In Proceedings of the High Capacity Optical Networks and Enabling Technologies, (HONET), Riyadh, Saudi Arabia, 19–21 December 2011; pp. 244–247.
13. Nizamuddin; Chudry, S.A.; Nasar, W.; Javaid, Q. Efficient signcryption schemes based on hyper elliptic curve cryptosystem. In Proceedings of the 7th International Conference on Emerging Technologies (ICET), Islamabad, Pakistan, 5–6 September 2011; pp. 1–4.
14. Malone-Lee, J. Identity Based Signcryption. Cryptology ePrint Archive, Report 2002/098. 2002. Available online: https://eprint.iacr.org/2002/098.pdf (accessed on 26 May 2019).
15. Libert, B.; Quisquater, J.J. A new identity based signcryption scheme from pairings. In Proceedings of the IEEE Information Theory Workshop (Cat. No.03EX674), Paris, France, 31 March–4 April 2003. [CrossRef]
16. Chow, S.S.M.; Yiu, S.M.; Hui, L.C.K.; Chow, K.P. Efficient Forward and Provably Secure ID-Based Signcryption Scheme with Public Verifiability and Public Ciphertext Authenticity. *Lect. Notes Comput. Sci.* **2004**, 352–369. [CrossRef]

17. Boyen, X. Multipurpose Identity-Based Signcryption. *Lect. Notes Comput. Sci.* **2003**, 383–399. [CrossRef]
18. Chen, L.; Malone-Lee, J. Improved identity-based signcryption. In Proceedings of the Public Key Cryptography-PKC'05, LNCS 3386, Les Diablerets, Switzerland, 23–26 January 2005; pp. 362–379.
19. Barreto, P.S.L.M.; Libert, B.; McCullagh, N.; Quisquater, J.-J. Efficient and Provably-Secure Identity-Based Signatures and Signcryption from Bilinear Maps. In *Advances in Cryptology—ASIACRYPT 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 515–532.
20. Selvi, S.S.D.; Vivek, S.S.; Rangan, C.P. Identity based public verifiable signcryption scheme. In Proceedings of the ProvSec'10, LNCS 6402, Malacca, Malaysia, 13–15 October 2010; pp. 244–260.
21. Yu, Y.; Yang, B.; Sun, Y.; Zhu, S. Identity based signcryption scheme without random oracles. *Comput. Stand. Interfaces* **2009**, *31*, 56–62. [CrossRef]
22. Jin, Z.; Wen, Q.; Du, H. An improved semantically-secure identity-based signcryption scheme in the standard model. *Comput. Electr. Eng.* **2010**, *36*, 545–552. [CrossRef]
23. Li, F.; Muhaya, F.B.; Zhang, M.; Takagi, T. Efficient Identity-Based Signcryption in the Standard Model. *Lect. Notes Comput. Sci.* **2011**, 120–137. [CrossRef]
24. Liu, Z.; Hu, Y.; Zhang, X.; Ma, H. Certificateless signcryption scheme in the standard model. *Inf. Sci.* **2010**, *180*, 452–464. [CrossRef]
25. Miao, S.; Zhang, F.; Li, S.; Mu, Y. On security of a certificateless signcryption scheme. *Inf. Sci.* **2013**, *232*, 475–481. [CrossRef]
26. Weng, J.; Yao, G.; Deng, R.; Chen, M.; Li, X. Cryptanalysis of a certificateless signcryption scheme in the standard model. *Inf. Sci.* **2011**, *181*, 661–667. [CrossRef]
27. Zhu, H.; Li, H.; Wang, Y. Certificateless signcryption scheme without pairing. *J. Comput. Res. Dev.* **2010**, *47*, 1587–1594.
28. Huang, Y.; Zhang, J.; Chen, H. On the Security of a Certificateless Signcryption Scheme. In Proceedings of the IWECA 2014, Ottawa, ON, Canada, 8–9 May 2014; pp. 664–667.
29. Shi, W.; Kumar, N.; Gong, P.; Zhang, Z. Cryptanalysis and improvement of a certificateless signcryption scheme without bilinear pairing. *Front. Comput. Sci.* **2014**, *8*, 656–666. [CrossRef]
30. Cheng, L.; Wen, Q. An improved certificateless signcryption in the standard model. *Int. J. Netw. Secur.* **2015**, *17*, 597–606.
31. Islam, S.; Li, F. Leakage-free and provably secure certificateless signcryption scheme using bilinear pairings. *Comput. J.* **2015**, *58*, 2636–2648. [CrossRef]
32. Li, L.; Shirase, M.; Takagib, T. Certificateless hybrid signcryption. *Math. Comput. Mod.* **2013**, *57*, 324–343. [CrossRef]
33. Mohapatra, R.K. Signcryption Schemes with forward Secrecy Based on Elliptic Curve Cryptography. Ph.D. Thesis, 2010. Available online: http://ethesis.nitrkl.ac.in/2009/ (accessed on 26 May 2019).
34. Elkamchouchi, H.M.; El-Atiky, M.H.; Abouelkheir, E. A Public Verifiability Signcryption Scheme without Pairings. *Int. J. Comput. Appl.* **2017**, *157*, 35–40.
35. Vigano, L. Automated security protocol analysis with the AVISPA tool. *Electron. Notes Theor. Comput. Sci.* **2006**, *155*, 61–86. [CrossRef]
36. Dolev, D.; Yao, A. On the Security of Public-Key Protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
37. Basin, D.; Modersheim, S.; Vigan, L. An On-The-Fly Model-Checker for Security Protocol Analysis. In *Proceedings of the ESORICS'03*; Springer-Verlag: Berlin/Heidelberg, Germany, 2003; LNCS 2808; pp. 253–270.
38. Clark, J.; Jacob, J. A Survey of Authentication Protocol Literature: Version 1.0. 17 November 1997. Available online: www.cs.york.ac.uk/~{}jac/papers/drareview.ps.gz (accessed on 26 May 2019).
39. Donovan, B.; Norris, P.; Lowe, G. Analyzing a Library of Security Protocols using Casper and FDR. In Proceedings of the FLOC'99 Workshop on Formal Methods and Security Protocols (FMSP'99); 1999. Available online: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.7256 (accessed on 26 May 2019).
40. Chevalier, Y.; Vigneron, L. Automated Unbounded Verification of Security Protocols. In *Proc. CAV'02, LNCS 2404*; Springer: Berlin/Heidelberg, Germany, 2002; Available online: https://link.springer.com/chapter/10.1007/3-540-45657-0_24 (accessed on 26 May 2019).

41. Armando, A.; Compagna, L. SATMC: A SAT-based Model Checker for Security Protocols. In *Proc. JELIA'04, LNAI 3229*; Springer: Berlin/Heidelberg, Germany, 2004; Available online: https://link.springer.com/chapter/10.1007/978-3-540-30227-8_68 (accessed on 26 May 2019).
42. Boichut, Y.; Heam, P.-C.; Kouchnarenko, O.; Oehl, F. Improvements on the Genet And Klay Technique to Automatically Verify Security Protocols. In Proc. AVIS'04, ENTCS. Available online: https://www.researchgate.net/publication/246435265_Improvements_on_the_Genet_and_Klay_technique_to_automatically_verify_security_protocols (accessed on 26 May 2019).
43. Ullah, S.; Junaid, M.; Habib, F.; Sana; Insafullah; Hizbullah. A novel proxy blind signcryption scheme based on hyper elliptic curve. In Proceedings of the 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, China, 13–15 August 2016. [CrossRef]
44. Zhou, C.; Zhao, Z.; Zhou, W.; Mei, Y. Certificateless Key-Insulated Generalized Signcryption Scheme without Bilinear Pairings. *Secur. Commun. Netw.* **2017**, *2017*, 8405879. [CrossRef]
45. Ch, S.A.; Nizamuddin; Sher, M. Public Verifiable Signcryption Schemes with Forward Secrecy Based on Hyperelliptic Curve Cryptosystem. *Commun. Comput. Inf. Sci.* **2012**, 135–142. [CrossRef]
46. Chaudhry, S.A.; Farash, M.S.; Naqvi, H.; Sher, M. A secure and efficient authenticated encryption for electronic payment systems using elliptic curve cryptography. *Electron. Commer. Res.* **2015**, *16*, 113–139. [CrossRef]
47. Yang, J.H.; Chang, Y.F.; Chen, Y.H. An efficient authenticated encryption scheme based on ECC and its application for electronic payment. *Inf. Technol. Control* **2013**, *42*, 315–324. [CrossRef]