Journal of Intelligent & Fuzzy Systems xx (20xx) x–xx
DOI:10.3233/JIFS-189166
IOS Press

An hyper elliptic curve based efficient signcryption scheme for user authentication

Malathi Devarajan* and N. Sasikaladevi
School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India

Abstract. With ever growing popularity, wireless communication system also vulnerable to various security attacks. To provide high level security, many cryptographic solutions have been proposed. One such solution is signcryption, where authenticity and confidentiality provided by single logical step. Therefore, signcryption scheme helps to reduce computational cost, but it is not feasible for resource constraint environments. Because, most of the existing approaches were based on El-Gamal, bilinear pairing, Rivest-Shamir-Adleman (RSA), and Elliptic curve Cryptography (ECC). They consume more energy due to their increased key size. Hence, the new signcryption approach is proposed based on Hyper Elliptic Curve Cryptosystem (HECC) whose key size is much lesser than ECC. It significantly reduces the cost of computation and communication overhead by half the amount of ECC which suits well for resource constraint environments. Further, the proposed scheme attains necessary security features along with forward secrecy and public verifiability. On the other hand, the security of the approach is validated through an automated protocol validation tool – AVISPA.

Keywords: Hyper elliptic curve, signcryption, mutual authentication, security analysis, AVISPA

1. Introduction

With the speedy advancement of communication system and its transition from wired to wireless network, many real-time applications were developed based on wireless technology. But applications like remote healthcare, field monitoring, e-voting, e-payment, banking and e-commerce shares highly sensitive information. Providing security for such sensitive information while communicating over unsecure channel gradually becomes a major challenge. Moreover, these applications depend mostly on resource constraint environment for processing and service delivering. Therefore performing heavy cryptography operations may consume more energy, space storage and other resources which do not suit well for resource constraint environments. It necessitates the development of new method which achieves necessary security requirements like non-repudiation, confidentiality, integrity and authenticity along with reduced computational and communication overhead [1–3].

In earlier days, various cryptosystems were used to ensure message confidentiality. As time passes, cryptosystem were generally categorized into symmetric and asymmetric cryptosystem. Symmetric cryptosystems are fast and easy to implement, hence it can be applicable to resource constraint environments. Whereas asymmetric cryptosystems are slow and also consumes more energy but provides digital signature. With asymmetric cryptosystem, user authentication became possible [4]. To ensure secured authenticated communication, "signature-then-encryption" method was used earlier. It simultaneously generates signature and performs encryption before transmitting the message. It significantly achieves cryptographic properties more efficiently than traditional approaches [5]. But "signature-then-encryption"

*Corresponding author. Malathi Devarajan, School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India. E-mail: malathipuducherry@gmail.com.

method consumes more energy and double folds the requirements of resources as it performs two distinct operations. They are also computationally complex.

Fortunately, Zheng introduces the concept called signcryption in 1997 to accomplish user authentication and message confidentiality at single step [6]. Signcryption is the method of performing both encryption and signature generation in one step to reduce communication overhead and computational cost of the system. At present, many signcryption methods have been proposed on the basis of different cryptosystems such as Rivest-Shamir-Adleman, Bilinear Pairing, El-Gamal, and Elliptic curve cryptosystems. But they are also computationally complex and security of such schemes is not fully verified with valid tools or proof theory. Besides, computational complexity of existing signcryption schemes impedes its application on real-time problems [7]. It necessitates the improvement of existing signcryption schemes to achieve desired security requirements in addition with reduced computational cost, storage space and bandwidth.

1.1. The key contributions

The key contributions of the proposed work are given as follows

- An improved signcryption scheme is proposed using hyper elliptic curve cryptosystem to reduce computational cost.
- The security of the proposed approach is validated using an automated simulation tool – AVISPA and proved that the protocol satisfies defined goals.
- Further, security analyses were made informally to demonstrate the achievement of necessary security requirements
- Finally, cost analysis were made and compared with other existing methods.

The remainder of the work is organized as follows: At first, existing signcryption schemes and its limitations are analyzed. Then essential mathematical background is presented for clear understanding of the proposed scheme. In section 4, proposed signcryption method is explained together with correctness. Next, the protocol is implemented on the simulation tool –AVISPA for protocol validation. Followed by informal security analysis and cost estimation. The paper finally concludes the work with future directions.

2. Related work

As an alternative to signature-then-encryption approach, signcryption method was introduced in 1997 by Zheng [8] to reduce communication overhead and computational cost of the former method. Later, Zheng and Imai [9] modified the Zheng's scheme with elliptic curves to provide security equivalent to RSA and El-Gamal cryptosystem but uses smaller key size. It reduces the cost of communication by 58% and computation by 40% of earlier signature-then-encryption algorithm which is based on EC cryptosystem. Though their scheme provides high security, it lacks forward secrecy and public verifiability.

To conquer the limitations of Zheng and Imai's scheme, Hwang et al. [10] presented ECC based signcryption algorithm. It achieves desired security requirements together with forward secrecy and public verifiability. The reduced computational cost produced by the scheme makes the scheme suitable for applications with resource constraints. Later Toorani and Beheshti [11] presents signcryption scheme using ECC to ensure forward secrecy. But to support public verifiability, user has to disclose their session key to trusted authority for verification. Another ECC based signcryption scheme is presented by Bala et al. [12] for key management. Though it supports forward secrecy, the scheme suffers from public verifiability. After that many signcryption schemes have been proposed with different cryptosystems.

In the present study, HECC were explored to design a signcryption scheme. HEC is the generalized figure of EC [13], but cryptosystem based on HEC provides equal security of ECC but with smaller base field and hence suits well for resource constraint environments. Their security relies on the complexity of breaking HEC discrete logarithmic problem and achieves same degree of security than other public key cryptosystems but with lesser key size, storage space, system overhead and communication bandwidth. The NIST recommended key size of HEC cryptosystem to produce equal degree of security provided by RSA with 1024 bit and ECC with 160 bit key size is 80 bit [14].

Further, to reduce the communication overhead and storage space, Ch and Amin [15] utilized HEC cryptosystem to construct the signcryption scheme with forward secrecy which suits well for resource constraint applications. Then Ch and Sher [16] improved the previously proposed signcryption scheme and presented the new public verifiable

signcryption scheme [17] using HECC. Motivated by the objective, the paper aims to propose new siggncryption scheme with users' aadhar number and HEC cryptosystem.

The added advantage of the proposed signcryption scheme is it ensures both forward secrecy and public verifiability together with reduced computational cost and storage space. The term forward secrecy refers that though the sender's private key is disclosed, the future session key will not be affected. Similarly, public verifiability refers to if any disputes arise, then trusted authority can resolve the problem without revealing the secret parameters. The efficiency and superiority of proposed signcryption method is analyzed and comparison is performed with previous methods in terms of computational cost.

3. Technical background

In this section, the mathematical background of hyper elliptic curve cryptosystem, definition of signcryption and an automated validation tool used to verify the proposed scheme is given for clear understanding of the scheme.

3.1. Hyper elliptic curve cryptosystem (HECC)

Koblitz, N. and Miller, V. first introduced elliptic curves into asymmetric cryptosystem in 1985. The security of ECC relies on the complexity of breaking discrete logarithmic problem (DLP) on points of elliptic curves. Elliptic curve with genus greater than one is defined as hyper elliptic curve. In other words hyper elliptic curve with genus one is defined as elliptic curve. And hence hyper elliptic curve is the extension of elliptic curve. Like ECC, the security of HEC cryptosystem also relies on the complexity of breaking discrete logarithmic problem in the jacobian curve. Figure 1 depicts the Genus-2 Hyper Elliptic Curve.

Let C be a hyper elliptic curve and F_q be finite field, where q is the large prime number, then C is described as follows

$$y^2 + h(x)\, y = f(x)$$

where $f(x)$ is a monic polynomial of degree less than or equal to $2g+1$, and $h(x)$ is a polynomial of degree less than or equal to g. Points on EC forms a group but that does not possible for HEC points to form a group. So a Jacobian group $J(F_q)$ is created with

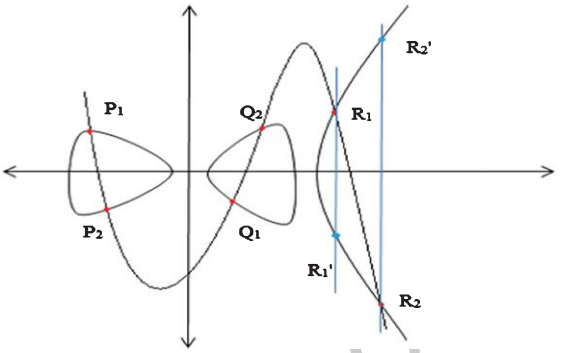

Fig. 1. Hyper Elliptic Curve of Genus 2.

HEC points and a point at infinity. Then a divisor D is selected from $J(F_q)$ which is also the generator of $J(F_q)$. Divisor D is the official sum of HEC points which can be symbolized in the Mumford form as

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

The group operations of HEC are divisor addition and divisor doubling. Let D_1 be divisor and $k \in F_q$ be an interger, then finding k from D_2 such that $D_2 = k * D_1$ is computationally complex, though D_1 and D_2 is known. It is defined as HEC discrete logarithmic problem [18]. HEC cryptosystems are superior over other public key cryptosystems due to its shorter key size and able to provide equal level security.

3.2. Signcryption

Signcryption is a primitive cryptographic function used to provide confidentiality and authentication for messages transmitting over unsecure channel. It simultaneously performs signature and encryption in single step. It was first proposed by Y. Zheng in 1997. After that many signcryption algorithms have been introduced based on different cryptographic techniques. But signcryption scheme with elliptic curve cryptosystems ensures high security with less key size and hence it subsequently reduces the storage space and communication cost.

In recent years, the extended version of elliptic curve i.e. hyper elliptic curve is used to design signcryption schemes. The advantage of using HECC for signcrytion scheme is it further reduces the memory size and other resources. So that it suits well for resource constraint applications like e-payment, e-commerce, banking, etc. In addition HECC based

signcryption method supports public verifiability, forward secrecy and other security necessities like confidentiality, non-repudiation, integrity and authentication [19, 20].

3.3. AVISPA: A simulation tool for protocol validation

The proposed signcryption scheme is validated using the simulation tool – AVISPA [21, 22]. AVISPA is a well known automated simulation tool used for internet security protocol verification. With respect to security parameters, it says whether the protocol is unsafe or safe. In order to obtain the simulation results, the protocol is coded in High Level Protocol Specification Language (HLPSL). Then the coded protocol is translated into low-level language via intermediate format (HLPSL2IF). Then one of the four back end checker is executed to validate the protocol [23, 24]. Then the results are displayed based on the selected back-end checker. The four back-end checkers are as follows:

- **Constraint-Logic based Attack Searcher (CL-AtSe):** It applies constraints logic on the protocol to detect flaws. CL-AtSe translates the code into set of constraints to find the possible attacks on the protocol.
- **On-the-Fly Model Checker (OFMC):** OFMC employed to detect the attacks presented on the protocol quickly and also verifies it in a demand driven way for bounded number of sessions.
- **Tree Automata for Security Protocol Analysis (TA4SP):** It approximates (under or over approximation) the knowledge of an intruder in tree language and presents whether it is flawed or safe for unbounded number of sessions.
- **SAT-based Model Checker (SATMC):** It creates a propositional formula and fed into State-of-the-Art (SAT) solver to find an attack.

In the HLPSL code, roles are specified for all participating agents. The header contains the information about each agent, symmetric or public key used, and send/receive channel. It follows the Dolev-Yav model. Then local variables are defined with secret parameters and authentication channel. Finally in the role session, parameters needed to start the session are given. And in the role environment, along with intruder knowledge, goals to achieve should be mentioned. As a result, the tool outputs the simulated result as either safe or unsafe or the attacks found or reason for not outputting the proper results.

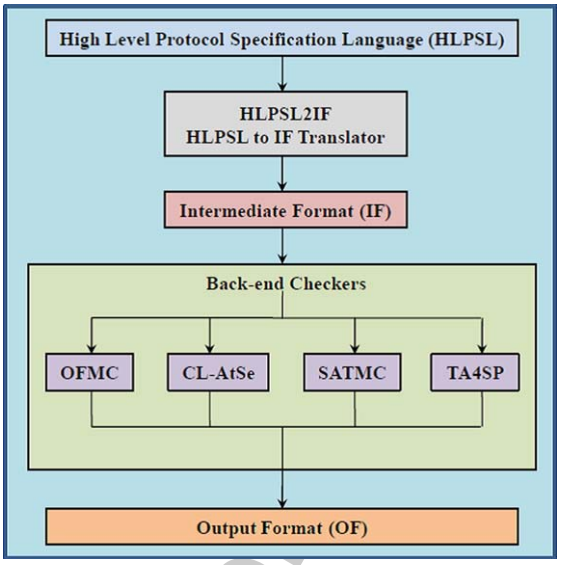

Fig. 2. Architecture of AVISPA – a protocol validation tool.

Table 1
Symbols used in the proposed method

Notation	Definition
C	Hyper elliptic curve over F_q
F_q	Finite field
q	Large prime number
D	Divisor
n	Random integer
AN	Aadhaar number of signcrypter
SK	Session key
d_s, d_u	Secret key of signcrypter and unsigncrypter
P_s, P_u	Public key of signcrypter and unsigncrypter
C	Ciphertext
(R, S)	Signature
T	Timestamp
$H()$	Hash function
E/D	Encryption/Decryption
M	Message to be sent

Figure 2 depicts the architecture of AVISPA – a security protocol validation tool.

4. The proposed signcryption method using HECC

An efficient signcryption method is designed using hyper elliptic curve cryptosystem to ensure message confidentiality and authenticity together with forward secrecy and public verifiability. It minimizes the cost of computation by half the amount of ECC based signcryption scheme due to shorter key size. Table 1 defines the symbols used in the method.

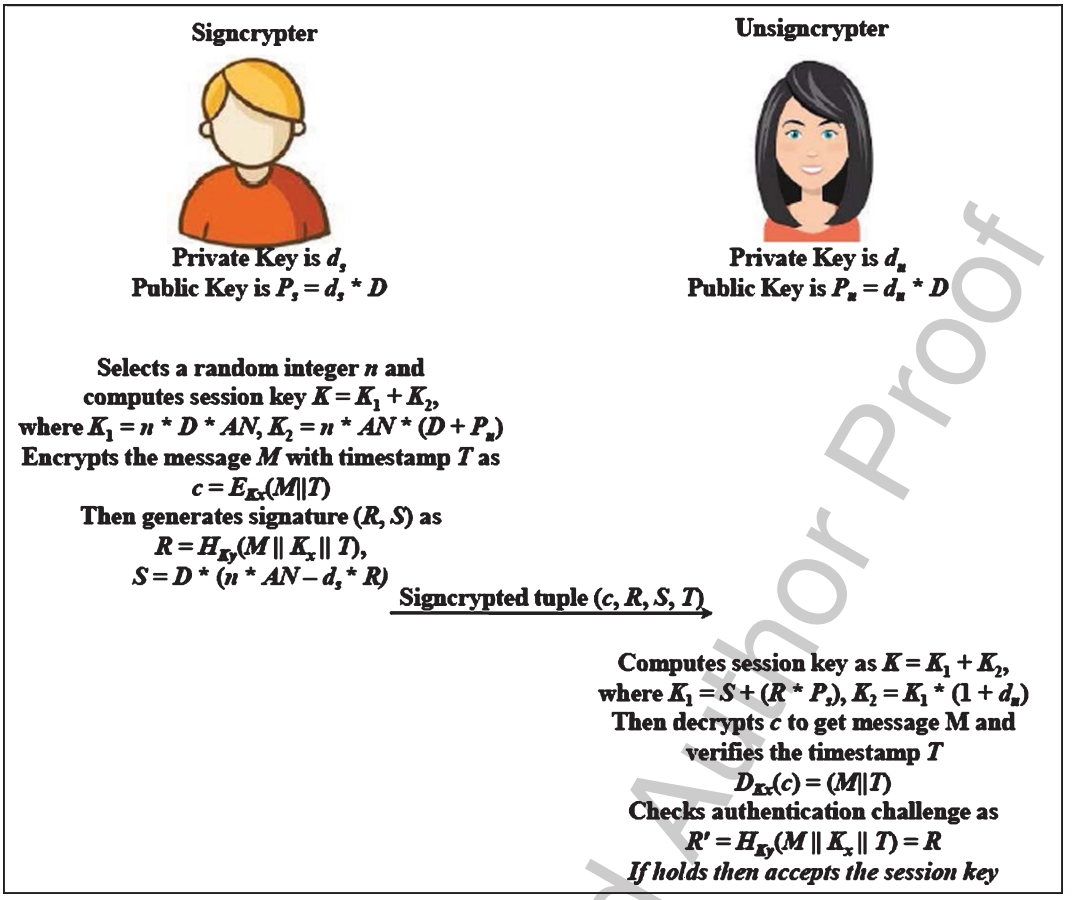

Fig. 3. Sequence of proposed Signcryption Scheme.

The three different phases of signcryption scheme are key generation, signcryption and unsigncryption phase. Explanation of each phase is given individually. Figure 3 depicts the steps followed to generate signcryption by sender and verification by receiver.

4.1. Key generation phase

Initially private keys are chosen by each user and its corresponding public keys are computed and then public keys are published globally along with public parameters in the key generation phase such as hyper elliptic curve C, divisor D, encryption/decryption algorithm and one-way hash function. Suppose $d_s \in F_q$ be the private key of signcrypter, then public key is $P_s = d_s{}^*D$. Similarly, private and public key of unsigncrypter is $d_u \in F_q$ and $P_u = d_u{}^*D$.

4.2. Signcryption

The public keys are publicly available for all users. If the signcrypter wishes to send a message securely, first he gets the public key of corresponding receiver. Then the following steps are performed to create signcrypted tuple (c, R, S, T). For that, signcrypter needs his Aadhaar number (AN) and random integer $n \in F_q$ to generate the session key K. In the proposed signcryption, signcrypter aadhaar number is used to avoid malicious user.

Step 1. Signcrypter picks the random number $n \in \{1, \ldots, q-1\}$ and calculates $K = K_1 + K_2$, where $K_1 = n^*AN^*D$, and $K_2 = n^*AN^*(D + P_u)$.

Step 2. The mapping function ψ is used to convert the polynomial key K into its corresponding integer (K_x, K_y).

Step 3. Then the message m is encrypted with K_x as, $C = E_{k_x}(m, T)$ and signature (R, S) is generated as, $R = H_{k_y}(m \,||K_x||\, T)$ and $S = (n^*AN - d_s{}^*R)\ {}^*D$, where T is the timestamp.

Step 4. Finally, the signcrypted tuple (c, R, S, T) is sent to corresponding unsigncrypter for verification through unsecure channel.

4.3. Unsigncryption phase

In this phase, the received tuple is unsigncrypted for signature verification and user authentication.

Step 1. Now unsigncrypter computes the key $K = K_1 + K_2$, where $K_1 = S + (R^*P_s)$, $K_2 = K_1(1 + d_u)$.

Step 2. Then mapping function ψ is used to convert the polynomial divisor K into integer $(K_x,\ K_y)$.

Step 3. Now the message is retrieved from the ciphertext c as $D_{K_x}(c) = (m, T)$.

Step 4. If the timestamp obtained from ciphertext matches with received timestamp, then it is considered that the message is fresh and authentication challenge is verified as, $R' = H_{K_y}(m \parallel K_x \parallel T) = R$. If holds, then unsigncrypter believes that message is from valid user, else the session is ignored.

4.4. Verification

$$
\begin{aligned}
K_1 &= S + \left(R^*P_s\right) \\
&= \left(n^*AN - d_s{}^*R\right)\ {}^*D + R^*P_s \\
&= D^*n^*AN - D^*d_s{}^*R + R^*P_s \\
&= D^*n^*AN - R^*P_s + R^*P_s \\
&= D^*n^*AN \\
&= K_1
\end{aligned}
$$

$$
\begin{aligned}
K_2 &= K_1(1 + d_u) \\
&= D^*n^*AN(1 + d_u) \\
&= n^*AN\left(D + D^*d_u\right) \\
&= n^*AN(D + P_u) \\
&= K_2
\end{aligned}
$$

5. Protocol validation using AVISPA

In an aim to prove that proposed signcryption scheme resists various attack, the scheme is verified using an automated simulation tool. AVISPA is an automated tool employed for protocol validation and outputs weather the protocol is safe or not [25, 26]. The security protocol is implemented in the High Level Protocol Specification Language and converts it into intermediate format through HLPSLIF. And then executes any of the 4 back-end checkers, i.e. CL-AtSe checker, OFMC checker, SATMC checker and TA4SP checker for security analysis. And then outputs the analysis report.

The HLPSL code of the proposed scheme consists of four roles, one for signcrypter and other for unsigncrypter. Remaining two roles are session and environment that are mandatory for every protocol. Figs. 4 to 6 represent the HLPSL code of our proposed scheme and the variables used. The header of each role contains the global and local variables and send/receive channel. Here, the send/receive channel uses the Dolev-Yao attack model [27]. After seeing the start signal, the protocol starts executing the session. Since signcrypter initiates the communication, he only receives the start signal and then establishes the link with unsigncrypter securely.

After computing the session key K, signcrypter encrypts the message with that key and creates the signature (R, S), then sends the signcrypted tuple to unsigncrypter with timestamp. The keyword SND() is used to send the message securely. It is observed from the code that secret(N′, s1, Signcrypter), secret(AN, s2, Signcrypter), secret(Ds, s3, Signcrypter) and secret(K′, s4, Signcrypter, Unsigncrypter) helps to maintain the fresh nonce N, aadhaar number AN, private key Ds and encryption key K' to be secret.

On the other hand, unsigncrypter receives the signcrypted tuple using the keyword $RCV()$ and decrypts the ciphertext to acquire the secret message with the key K. The key is computed by unsigncrypter with the help of received signature (R, S) and his private key Du. And also verifies the signature, if holds then unsigncrypter accepts the message. Here, the private key Ds is kept secret. The statement witness(Signcrypter, Unsigncrypter, auth_a1, {M′}) is used for authentication purpose and declares that unsigncrypter is witness for the message M. These goals are mentioned in the goal section.

Finally, the role session and environment is defined with goals to achieve, intruder knowledge and sessions. And the back end checker is executed for

SPAN 1.6 - Protocol Verification : signcryption.hlpsl

File

```
role signcrypter(
        Signcrypter, Unsigncrypter:agent,
        Ps,Pu:public_key,
        Add,Sub,Mul,H:hash_func,
        SND,RCV:channel(dy))
played_by Signcrypter
 def=
        local
                State:nat,
                N,D,AN,T,M,Ds:text,
                K1,K2,C,R,S,S1,K:message
                const s1,s2,s3,s4,s5,auth_a1:protocol_id
        init
                State:=0
        transition
                1. State=0 /\ RCV(start)=|>  State':=1 /\ N':=new() /\ M':=new()
                /\ K1':=Mul(N'.D.AN)  /\ K2':=Mul(N'.AN.Add(D.Pu))  /\ K':=Add(K1'.K2')
                /\ C':={M'.T}_K'  /\ R':=H({M'.K'.T}_K')
                /\ S1':=Sub(Mul(N'.AN).Mul(Ds.R'))  /\ S':=Mul(D.S1')
                /\ SND({M'.T}_K',R'.S'.T)
                /\ secret({N'}, s1, {Signcrypter})  /\ secret({AN}, s2, {Signcrypter})
                /\ secret({Ds}, s3, {Signcrypter})  /\ secret({K'}, s4, {Signcrypter,Unsigncrypter})
end role
```

Fig. 4. HLPSL code for role signcrypter.

SPAN 1.6 - Protocol Verification : signcryption.hlpsl

File

```
role unsigncrypter(
        Signcrypter, Unsigncrypter:agent,
        Ps,Pu:public_key,
        Add,Sub,Mul,HMAC:hash_func,
        SND,RCV:channel(dy))
played_by Unsigncrypter
 def=
        local
                State:nat,
                Du,M,T:text,
                K1,K2,K,R,S:message
                const s1,s2,s3,s4,s5,auth_a1:protocol_id
        init
                State:=0
        transition
                1. State=0 /\ RCV({M'.T'}_K'.R'.S'.T)=|>
                State':=1
                /\ K1':=Add(S'.Mul(R'.Ps))
                /\K2':=Mul(K1'.Add(1.Du))
                /\K':=Add(K1'.K2')
                /\ secret({Du}, s5, {Unsigncrypter})
                /\witness(Signcrypter,Unsigncrypter,auth_a1,{M'})
end role
```

Fig. 5. HLPSL code for role unsigncrypter.

security analysis. The back-end checkers analysis the protocol and outputs the result with various sections. The section SUMMARY says whether the protocol is unsafe or safe or inconclusive. In DETAILS section, conditions on which the scheme is unsafe or why it is inconclusive is shown. The other sections such as PROTOCOL, GOAL, BACKEND, COMMENTS and STATISTICS states the protocol name, state of

SPAN 1.6 - Protocol Verification : signcryption.hlp

File

```
role session(
        Signcrypter, Unsigncrypter:agent,
        Ps,Pu:public_key,
        Add,Sub,Mul,HMAC:hash_func)
def=
        local
                SND1,RCV1,SND2,RCV2:channel(dy)
        composition
                signcrypter(Signcrypter,Unsigncrypter,Ps,Pu,Add,Sub,Mul,HMAC,SND1,RCV1)
                /\unsigncrypter(Signcrypter,Unsigncrypter,Ps,Pu,Add,Sub,Mul,HMAC,SND2,RCV2)
end role

role environment()
def=
        const
                signcrypter,unsigncrypter:agent,
                ps,pu:public_key,
                add,sub,mul,hmac:hash_func,
                s1,s2,s3,s4,s5,auth_a1:protocol_id
        intruder_knowledge={signcrypter,unsigncrypter,ps,pu,add,sub,mul,hmac}
        composition
                session(signcrypter,unsigncrypter,ps,pu,add,sub,mul,hmac)
                /\session(i,unsigncrypter,ps,pu,add,sub,mul,hmac)
                /\session(signcrypter,i,ps,pu,add,sub,mul,hmac)
end role

goal

authentication_on auth_a1
secrecy_of s1 secrecy_of s2 secrecy_of s3 secrecy_of s4 secrecy_of s5

end goal

environment()
```

Fig. 6. HLPSL code for role session and environment.

goal achievement, back-end used for analysis. If any attack is found, then sketch of an attack is presented together with comments and statistics.

To assess the security of proposed signcryption method, the protocol is implemented in HLPSL code and then the CL-AtSe and OFMC back-end checkers were executed. The simulation results of both the CL-AtSe and OFMC back-end checkers were gien in Figs. 7 and 8. From the simulation results, it is observed that our protocol is safe and achieves defined privacy goals.

SPAN 1.6 - Protocol Verification : sig

File

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/signcryption.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.06s
  visitedNodes: 4 nodes
  depth: 2 plies
```

Fig. 7. OFMC result.

6. Informal security analysis

The security requirements of wireless communication system are user authentication, message integrity, forward secrecy, confidentiality, unforgeability, non-repudiation, and public verifiability. The proposed HEC based signcryption method meets all the desired security features and its proof are explained individually in this section. The overall security of the proposed method depends on the complexity of solving HEC discrete logarithmic problem

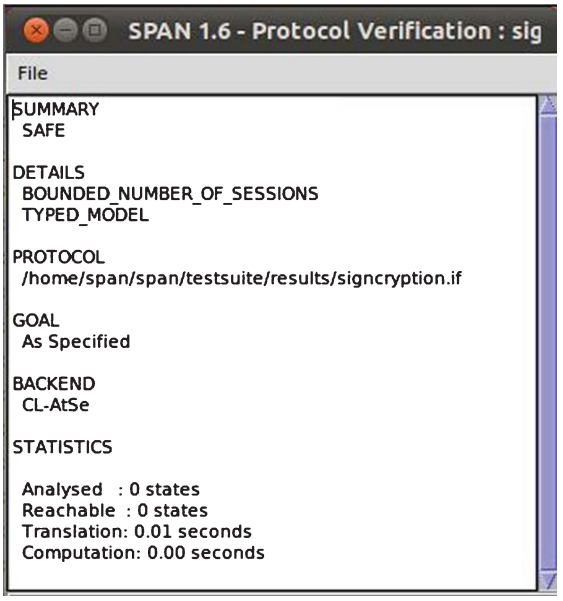

Fig. 8. CL-AtSe result.

which means it is computationally complex to find an integer n from D_2, where $D_2 = D_1 * n$, provided D_1 and D_2.

6.1. User authenticity

Our proposed signcryption scheme ensures user authenticity. As the session key (K_x, K_y) is generated by using senders' aadhaar number AN and signature by using private key d_s which is kept secret. Without knowing private key and aadhar number of sender, the signature cannot be generated and without session key, signcrypted tuple (c, R, S, T) cannot be constructed. Further, R is constructed by one-way hash function and hence any change in key or message provides inappropriate hash value. If received R doesn't matches with R', then the session is ignored by unsigncrypter.

6.2. Message confidentiality

The message to be communicated over unsecure channel should maintain confidentiality, because it may contain highly sensitive information. The proposed scheme ensures message confidentiality by means of symmetric encryption. It uses symmetric key K_x to encrypt the message. Confidentiality of the message can be compromised only an attacker can generate the exact symmetric key K_x. But it is difficult for an attacker to get the key K_x, because our scheme is protected by HEC discrete logarithm problem. Though an attacker knows the pubic key of unsigncrypter P_u, aadhar number of signcrypter AN and divisor D, it is computationally infeasible to obtain an integer n. Without knowing the random number n or private key d_u of unsigncrypter, computing the key (K_x, K_y) is infeasible.

6.3. Message integrity

The originality or reliability of the message received from signcrypter is verified by the unsigncrypter before the required action to be performed. If an attacker modifies the message, he has to modify the signature (R, S) in correspondence with altered ciphertext c'. But it is impossible to generate legitimate signature because it is computed by one-way hash function which is also a collision resistant. It also uses the session key (K_x, K_y) to produce hash value. As said before, in order to generate valid signature, an attacker has to solve HECDLP to get (K_x, K_y). Though the session key (K_x, K_y) is compromised, an attacker can generate R. But to generate S, he needs signcrypters' aadhar number AN, private key d_s and random integer n which is again computationally infeasible.

6.4. Unforgeability

The proposed scheme prevents the attacker from forging valid signcrypted tuple (c, R, S, T). To generate valid signature, he is in need of signcrypters' private key d_s and random number n. Since each session utilizes different random number, it is hard to guess private key and random number. And also from previous session keys, it is difficult to get the secret parameters, as he has to solve HECDLP. Hence, an attacker is unable to construct correct signature and thus our scheme achieves unforgeability.

6.5. Non-repudiation

The signcrypter cannot deny their transmitted message (c, R, S, T), because unsigncrypter has the facility to demonstrate that the message is received from corresponding signcrypter. If any conflicts arise between signcrypter and unsigncrypter, then trusted authority can involve and resolve the conflict. Therefore, our proposed method ensures non-repudiation.

Table 2
Comparative analysis of computational cost and execution time

Schemes	HECDM	HECDA	ECPM	ECPA	Mul/Div	Add/Sub	Hash	KH	Enc/Dec	Execution time
Hwang et al. [10]	–	–	5	1	1	1	2	–	2	5 *4.24 = 21.2 ms
Kumar et al. [28]	–	–	6	2	–	–	2	–	2	6 *4.24 = 25.44 ms
Ch et al. [17]	6	1	–	–	2	1	2	2	2	6 *2.2 = 13.2 ms
Heydari et al. [9]	7	1	–	–	1	1	–	2	2	7 *2.2 = 15.4 ms
Proposed	5	4	–	–	2	2	–	2	2	5 *2.2 = 11.0 ms

6.6. Forward secrecy

Though the secret key of signcrypter is intercepted by an adversary, the session key will not be disclosed. Here the symmetric key K_x is utilized for encryption, which cannot be generated by an attacker. Because, even though an attacker knows the aadhaar number of signcrypter, he is not able to get senders' private key. Without these secret parameters, an attacker cannot able to create symmetric key. For that, an attacker needs to solve complex HECDLP. By this way, the proposed scheme achieves forward secrecy.

6.7. Public verifiability

If suppose that signcrypter denies the transmission of his signature, then recipient can raise the dispute and solves it by trusted authority. He can prove that message is received from legitimate sender without disclosing the content of the message. Trusted authority can verify the signature and solves the dispute using zero-knowledge protocol.

6.8. Resists replay attack

The scheme resists replay attack with the help of timestamp T. The signature (R, S) and ciphertext c is generated with the timestamp. Thus if an attacker tries to send previous message to the unsigncrypter, the message is ignored by the unsigncrypter. Because, unsigncrypter will find that the timestamp is not fresh and rejects the session.

7. Cost estimation

The key requirement of any wireless communication system with resource constraint device is to provide high security with less computational cost and communication overhead. The computational and communication cost of the proposed scheme is analyzed and presented in comparison with other related schemes.

7.1. Computational cost analysis

The computational cost of the proposed scheme is analyzed in a PC having Intel Core i3-6100U processor with 2.3 GHz speed and 4GB RAM with Windows 10 operating system. An expensive arithmetic operation of elliptic curve cryptosystem is point multiplication (ECPM). Since hyper elliptic curve is the generalized form of elliptic curve and uses half of the key size, HEC divisor multiplication (HECDM) will consume more cost than other operations. According to [14], the time taken to complete single ECPM is 4.24 ms and 2.2 ms for HECDM. Thus the proposed scheme is compared with existing schemes in terms of arithmetic operations are given in Table 2. The arithmetic operations considered for estimating computational cost of the proposed scheme is given as follows:

- *Hash*: One way hash function
- *KH*: Keyed hash function
- *ECPM*: EC point multiplication
- *ECPA*: EC point addition
- *HECDM*: HEC divisor multiplication
- *HECDA*: HEC divisor addition
- *Enc/Dec*: Symmetric encryption/decryption
- *Mul/Div*: Multiplication and Division
- *Add/Sub*: Addition and subtraction

The cost of the proposed signcryption scheme and its comparison with other related schemes is given in the Table 2. From the Table 2 it is observed that the proposed method provides less computational cost and hence it suits well for resource constraint applications.

7.2. Communication cost analysis

The major concern of wireless communication technology is high bandwidth and hence the proposed signcryption scheme aims to lower the communication overhead by using HECC. The parameters chosen for constructing the scheme and the amount of data to be communicated over the channel will decide

the communication cost. Let us assume that

- $|hash(x)| = |KH(x)| \approx |q|$, where q is the prime number and $\geq 2^{160}$
- $|hash(x)| = |KH(x)| \approx 2|n|$, where n is the prime number and $\geq 2^{80}$
- $|c| = |m|$, where c is the ciphertext of signcryption scheme
- $\left|c'\right| = 2|D|$ if $|m| \leq |D|$,
- $\left|c'\right| \geq 2|m|$ if $|m| \geq |D|$
- $D = (u(x), v(x)) = \left\{ \sum_{i=0}^{g} x^i u_i \sum_{i=0}^{g-1} x^i v_i \right\}$, where $u_i, v_i \leq 2n \longrightarrow |D| \geq 2|n|$

The major concern of wireless communication technology is high bandwidth and hence the proposed signcryption scheme aims to lower the communication overhead by using HECC. The communication cost of proposed scheme is $|c| + |D| + |n|$.

8. Conclusion and future directions

The study proposes the hyper elliptic curve based signcryption scheme which suits well for resource constraint environment like mobile applications such as e-payment, e-banking, e-commerce, healthcare and field monitoring. The scheme meets all the necessary security requirements and also resists various cryptographic attacks. Furthermore, it consumes less computational cost and storage space when compared with other existing signcryption schemes. From the simulation results of AVISPA, it is concluded that the proposed signcryption scheme can find its role in many real-time wireless applications. Informal security analyses were also made against all possible attacks.

Acknowledgments

The authors are grateful to Science and Engineering Research Board (SERB), Department of Science and Technology, New Delhi, for the financial support under ECR grant (ECR/2017/000679/ES).

References

[1] W. Stallings, Network and internetwork security: principles and practice (Vol. **1**). Englewood Cliffs, NJ: Prentice Hall, (1995).

[2] S. Ullah, X.Y. Li and L. Zhang, A review of signcryption schemes based on hyper elliptic curve, In *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)* (pp. 51–58). IEEE, (2017, August).

[3] A. Mehmood, I. Ullah and A.I.U. Noor-Ul-Amin, Public Verifiable Generalized Authenticated Encryption (ℙℙ𝔾 Æ) based on Hyper Elliptic Curve, *J Appl Environ Biol Sci* **7**(12) (2017), 69–73.

[4] S.A. Ch, W. Nasar and Q. Javaid, Efficient signcryption schemes based on hyperelliptic curve cryptosystem. In *2011 7th International Conference on Emerging Technologies* (pp. 1–4). IEEE, (2011, September).

[5] A. Braeken and P. Porambage, ASEC: anonym signcryption scheme based on EC operations, *International Journal of Computer Applications* **5**(7) (2015), 90–96.

[6] Y. Zheng, Signcryption and its applications in efficient public key solutions, *International Workshop on Information Security*, Springer, Berlin, Heidelberg, 1997.

[7] X.W. Zhou, Improved Signcryption Schemes Based on Hyper-elliptic Curves Cryptosystem, In *Applied Mechanics and Materials* (Vol. **20**, pp. 546–552). Trans Tech Publications Ltd. (2010).

[8] Y. Zheng, Digital signcryption or how to achieve cost (signature & encryption)« cost (signature)+cost (encryption). In *Annual international cryptology conference* (pp. 165–179). Springer, Berlin, Heidelberg, (1997, August).

[9] Y. Zheng and H. Imai, How to construct efficient signcryption schemes on elliptic curves, *Information processing letters* **68**(5) (1998), 227–233.

[10] R.J. Hwang, C.H. Lai and F.F. Su, An efficient signcryption scheme with forward secrecy based on elliptic curve, *Applied Mathematics and computation* **167**(2) (2005), 870–881.

[11] M. Toorani and A.A. Beheshti, An elliptic curve-based signcryption scheme with forward secrecy, *arXiv preprint arXiv:1005.1856*, (2010).

[12] S. Bala, G. Sharma and A.K. Verma, An improved forward secure elliptic curve signcryption key management scheme for wireless sensor networks, In *IT Convergence and Security 2012* (pp. 141–149). Springer, Dordrecht, (2013).

[13] N. Koblitz, Hyperelliptic cryptosystems, *Journal of cryptology* **1**(3) (1989), 139–150.

[14] R. Ganesan, M. Gobi and K. Vivekanandan, A Novel Digital Envelope Approach for A Secure E-Commerce Channel, *IJ Network Security* **11**(3) (2010), 121–127.

[15] S.A. Ch and N. Amin, Signcryption schemes with forward secrecy based on hyperelliptic curve cryptosystem, In *8th International Conference on High-capacity Optical Networks and Emerging Technologies* (pp. 244–247). IEEE, (2011, December).

[16] S.A. Ch and M. Sher, Public verifiable signcryption schemes with forward secrecy based on hyperelliptic curve cryptosystem. In *International Conference on Information Systems, Technology and Management* (pp. 135–142). Springer, Berlin, Heidelberg, (2012, March).

[17] S.A. Ch, M. Sher, A. Ghani, H. Naqvi and A. Irshad, An efficient signcryption scheme with forward secrecy and public verifiability based on hyper elliptic curve cryptography, *Multimedia Tools and Applications* **74**(5) (2015), 1711–1723.

[18] A. Sadat, R. Ahmad, I. Ullah, H. Khattak and S. Ullah, Multi Receiver Signcryption Based On Hyper Elliptic Curve Cryptosystem, *J Appl Environ Biol Sci* **7**(12) (2017), 194–200.

[19] A.K. Singh, A review of elliptic curve based signcryption schemes, *International Journal of Computer Applications* **102**(6) (2014).

[20] W. Stallings, Cryptography and network security: principles and practice, international edition: principles and practice. Pearson Higher Ed., (2014).

[21] L. Viganò, Automated security protocol analysis with the AVISPA tool, *Electronic Notes in Theoretical Computer Science* **155** (2006), 61–86.

[22] I. Ullah, N.U. Amin, M. Naeem, H. Khattak, S.J. Khattak and H. Ali, A Novel Provable Secured Signcryption Scheme PSSS: A Hyper-Elliptic Curve-Based Approach, *Mathematics* **7**(8) (2019), 686.

[23] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar,... and S. Mödersheim, The AVISPA tool for the automated validation of internet security protocols and applications, In *International conference on computer aided verification* (pp. 281–285). Springer, Berlin, Heidelberg, (2005, July).

[24] H.L.P.S.L. Tutorial, A Beginner's Guide to Modelling and Analysing Internet Security Protocols (2005). Available at [AH-03], (2009).

[25] T.A. Team, AVISPA v1. 1 User manual. Information society technologies programme (June 2006), http://avispa-project.org. (2006).

[26] L. Takkinen, Analysing security protocols with AVISPA, In *TKK T-110.7290 research seminar on network security*, (2006).

[27] D. Dolev and A. Yao, On the security of public key protocols, *IEEE Transactions on information theory* **29**(2) (1983), 198–208.

[28] R. Kumar, S.K. Pal and A. Yadav, Elliptic curve based authenticated encryption scheme and its application for electronic payment system, *International Journal of Computing Science and Mathematics* **9**(1) (2018), 90–101.