International Journal of Security and Its Applications
Vol. 10, No. 7 (2016) pp.287-306
http://dx.doi.org/10.14257/ijsia.2016.10.7.26

Effective Signcryption Approach for Secure Convention for Multilayer Consensus using ECC

Gautam Kumar and Hemraj Saini

Department of Computer Science & Engineering
Jaypee University of Information Technology
Solan, H.P.-173234 INDIA
Department of Computer Science & Engineering
Jaypee University of Information Technology
Solan, H.P.-173234 INDIA
gautam.kumar@mail.juit.ac.in, hemraj1977@yahoo.co.in

Abstract

The used algorithm in cryptography represents the facts for compution and/or computation costs in general. The motivation for any problem is primitive generators that makes the protocol a big advantage over the technology augmentation. This manuscript presents a methodological approach on session specific challenge-response protocol for a better, improved and stronger security on reduced costs. The basic primitives are applying on Diffie-Hellman and Elliptic Curve Cryptography. The purpose is proving the security properties for protocol compositional logic that focuses on privacy rights in information assessment in multidisciplinary obligations. In addition, we portrats a signcryption approach for password authenticated key exchange protocol for multilayer consensus, which logically combines individual signature and encryption cost in the form of reduced computational cost and communications cost in single stride of operation. The overall computation time potentially reduces for the proposed methodology on key generation and signature. The results for ECC based multilayer consensus key generation approach are testing on Automated Validation of Internet Security Protocol Architecture (AVISPA) tool and SPAN tool. Further, by preserving the definition of signcryption, we enhanced the same scheme in relations to the other proposed schemes.

Keywords*: ECC, Secure Composition, Challenge-Response, Signcryption, PDS, MCEPAK*

1. Introduction

The challenge-response (C-R) protocols are the recent research trends in cryptography, in this the logic has designed around a process calculus. It is including the actions to generate new random numbers, perform encryption and signature, send or/and receive messages, finally performing decryption with verification with matched digital signature. The security proofs allow the protocols to build the protocols by combining by independent proofs their parts. Secure composition as it designs in such a way that may not degrade and does not affect its own existing security so it has considered itself as a difficult security problem. This philosophy is more amenable for automation of security protocol analysis, where the cryptography has assumed to be perfect with respect to the speed, computational cost and communication cost, energy minimization, applicability for respective short-memory devices, *etc*. The goal is to develop protocol derivational system approach on behalf of logical methods, where the protocol analysis concernes to the soundness theorems. Datta [1] has presented an innovative framework for secure composition on its formal methods such as: *Protocol Composition Logic (PCL)* and

ISSN: 1738-9976 IJSIA

Protocol Derivation System (PDS). PDS is supporting the syntactic derivations of complex protocols with starting from basic components and combines or extends them using a sequence of operations like refinements, transformations and its compositions. Floyd-Hoare logic is a foundation of PCL that supports axiomatic proofs for protocol properties [2]. The PCL objective is to form a proof method for every applicable derivation for PDS. Therefore it may also be enable to its security proofs and may also be apply for parallel development of others protocols [3]. Any protocol execution contains as assertions associated with the same. The powerful possible observation offers reason to all runs of the protocol without any reasoning. The basic operation for ISO-9798-3 based on Diffie-Hellman exponential as (CR) protocol considers, as shown in Figure 1, that represents to show how messages are sent by one may be received by other. The basis of execution protocol consist on initiator role and responder role. The Initiator principle role is executing to generate a fresh random number, send the message with its generators to peer; the Responder receive message with source address of the peer(Sender); verify the same message that contains the signature in the anticipated format, and at the end both should be ready to send a subsequent another messages with signature of initiator and responder [4].

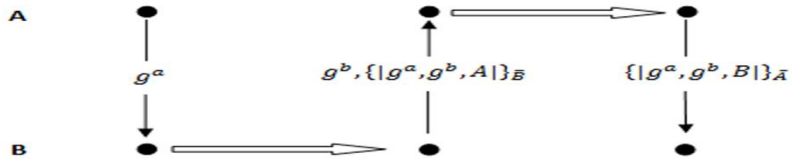


Figure 1. ISO-9798-3 Protocol

The backbone of security protocol is the foundational basis that is making certain to forward correctness in many distributed systems to error prone. In relation to presented protocols that contain security redundancies or flaws in the literature of subsequent sections. A simple logic is described as a consequence of communication and its progression towards the authenticated trustworthy parties involved in authentication protocols. Further, we consequently explained our proposed work formally for a variety of protocol families that ascertain the errors and naunces, and instead of that suggest improvements in them.

We have given a brief idea with intensification of Elliptic Curve Cryptography (ECC) in the next section, which consideres multilayer consensus key generations using the same. Then after, signcryption based approach considers which reduces the computational cost and communications cost on the C-R protocol and encryption cost in a single logical step.

2. Elliptic Curve Cryptography

The heart of cryptography is the Discrete Logarithmic Problem (DLP), which acts like a fundamental role of information security. The exciting feature with greater significance and high computational speed at a lower cost is always a demanding issue. For efficient implementation of cryptographic protocols are playing a central role for the same. Cryptographic algorithms used in the approach with the slow running approach impinged on customer dissatisfaction and inconvenience. The reason being is clear to say in the growing field of computation and communication security with faster running algorithms are leading with high speed and high performance.

Elliptic Curve Cryptography (ECC) [5] proposed in 1985 by Neal Koblitz and Victor Miller. Due to the rapid growth in the recent state of affairs other than the ECC algorithms are considered to be secure, but they require a much higher length key. For reason in increased length, the computational and/or communications cost involved in the

method is not better suited for low memory devices. ECC is one of the techniques that are used to provide the same level of security with comparatively very shorter key sizes. But, the current computational complexity is still in excess and improvements in the same are our motivating issue. The same cost can be minimized using parallelization with the new advanced approaches in overall consideration, has also been one of our motivations [6].

The core building block of the public key cryptography for ECC-DLP is described on two elliptic points P and Q on the curve, to evaluate the value k, such that $Q = kP$ [7]. This procedure restrains to itself as a repeated point addition (ADD) and point doubling (DBL) operations.

It has been considered an effective alternative approach relative to RSA algorithm of an already established due to some certain reasons like- low entropy random numbers, lack of forward secrecy, chosen cipher text attacks, higher complexity, the mathematical attacks, and higher computational cost due to higher key lengths. It also gives a guarantee to all the security services on shorter keys for ECC. Less arithmetic cost, time saving and less space for key storage are the special benefits, when keys are transmitting for the same. These characteristics are making ECC the best choice to provide security in mobile phones, smart cards, online banking, controllers like routers, consumer electronics, laser printers, bridges, applications in robotics, network devices, automotive, and many more. An increased ECC evidence are the inclusions in credited standards organizations for American National Standards Institute (ANSI), International Standards Organization (ISO), Institute of Electrical and Electronic Engineers (IEEE), and National Institute of Standards and Technology (NIST).

Hardware encryption devices are using for the cryptographic algorithms that are running on a general purpose physical security for the most operating systems. These devices are providing high speeds, high security and high performance services. Nowadays, mobile devices are using in the global communication world. To perform a specific task the devices are interacting with other devices and are forming ad-hoc networks. The aspects of key devices contain the security and interoperability in the heterogeneous inter-network environment. ECC is mostly best in use due to exhaustive scarce resources for mobile devices and prompts for feasible for high demands.

The recent research trends in cryptography, for research community, ECC has attracted the maximum attention in the last two-and-half decades and dominating RSA/DSA systems. The extension of ECC is reaching to better performance used in the cryptosystem because of an improved version of algorithm, the uses of special functionalities and specialized curves. It works on less memory and a much faster computation are the major thought. The requirements of memory size execution and code sizes are also smaller. Whenever the security problem is considered in more details, then the smaller input length can be used with sufficient security. ECC is appropriate algorithm that works on smaller key length and makes for an efficient practical applications. These are based on two problems such as prime and binary fields that are closely related to the key length for ECC. A large amount of published research are offering an interesting alternative to standard prime field based cryptosystems, the reason is clear to representing that for the same security level, it requires less memory and much faster performance. The elliptic curve scalar multiplication is evaluating (on general) by (1) equation-

$$\begin{cases} y^2 = (x^3 + ax + b) \bmod p \text{ where } (x, y) \epsilon \mathbb{Z}_p \\ (\text{for lagre prime field }) \; p > 3 \; a, b \epsilon \mathbb{Z} \text{ and } 4a^3 + 27b^2 \neq 0 \bmod p \end{cases} \quad (1)$$

The scalar multiplication used in the ECC is a backbone for every algorithms used in ECC primitives. These are principally based on three main approaches. The first one is based on prime or binary field operations in the underlying finite-field, and these are en-routing to itself as alternative replacements for better solutions. The second approach is the computation of scalar multiplication on behalf of the applied algorithms that decides

its complexity cost. In these regards, the various available algorithms are in the forms of Most Significant Bit first, Least Significant Bit first, Nonadjacent form (NAF), Window Method, Sliding Window Method, Width Nonadjacent Form [8], Frobenious Map [9], [10], [11] and Radix-rNAF (r-NAF) [12]. The final approach is to use more hardware support for utilizing memory, and/or parallelize operations [13]-[15], and/or pipelining methods [16] that decides its architecture behaviors. The presented manuscript combines the formal verification on the combination of the first two approaches, and for the parallel computation in relation to the third approach to yield an efficient scalar multiplication scheme.

The National Institute of Standard and Technology (NIST) [17] document specifies with an objective at pair wise key establishment by means of discrete logarithmic problems in cryptography and similar to auxiliary functions for other applications that promotes for alternative assumptions for applicability.

Further this text has organized from Section 3 to Section 4 into four (04) sections. An approach towards the protocol derivation has presented, in Section 3. In Section 4, Multilayer Consensus ECC-based password authenticated key exchange (PAKE) an auxiliary model has presented for standard ECC protocol. Section 5, depicts a formal verification using AVISPA & SPAN and finally in Section 6 presents applicability of signcryption with the enrichment to the applications.

3. Motivation Towards Derivation of Secure Protocol Composition

Protocol Derivation System (PDS) and Secure Composition of Protocol (PDS) address the two central problems under the security framework [18]. The goal is to developing methods for proving their security properties of complex protocols by identifying and/or combining the independent proofs of their independent parts. PDS supports logical derivations which starts from the basic component and/or extend or combine a sequence of component compositions, refinements and transformations operations. These are considering a list of basic building blocks elements, set of encryption operations which replaces a plaintext with an encrypted nonce, then transfer the same into specific channel and at the end it should be recovered unintelligible, respectively. A component consists of a set of roles such as server, an initiator and a responder, where each play a role of actions on desired protocol on a sequence of input, output operations. A common shared secrete key forms on behalf of the executing roles played by their own private signing keys on generated nonces. A Diffie-Hellman component C_1 is as an example that provides a way for two parties to set-up a shared key $g^{ir} \bmod q$ where passive attacker as usual can't be discover. Component C_2 has considered as signature-based authenticator at the other end as a challenge-response signature in the form of generated nonce [19]. The standard uthentication mechanism has shown below-

$$I \rightarrow R: m;\ R \rightarrow I: SIG_R(m) \tag{2}$$

In the cryptography, it has assumed that m is a fresh value or nonce. Public key certificate posses by responder R to verify its the signature using the transformation and refinements operations. The refinement shows the instance of a message component replaces by some other unidentifiable means, such kind of an action gives freshness guaranteed, guarantees from internet key exchanges, protection from identity representation against the passive attackers, forward secrecy etc. A basic thought of the existing refinements have presented here. For an identity protection, the first refinement $R_1, SIG_x(m) = E_k\big(SIG_x(m)\big)$ works, the second refinement $R_2, SIG_x(m) = SIG_x\big(HMAC_x(m, ID_x)\big)$, proves that the signature term is itself generated from x and in addition the key hash proves x possesses as key K. For Internet Key Exchange (IKE) this property is crucial for mutual authentication guarantee. The refinement $R_3, SIG_x(m) = SIG_x(m), HMAC_k(m, ID_x)$, this serves as a same purpose of R_2 instead of the same it

uses to derive for Just Fast Key protocol. Refinement R_4, $SIG_x(m) = SIG_x(m, ID_y)$, has assumed that x possess the required identifying information $Y's$ public key certificate, before the protocol executes. The refinement $R_5, g^x = g^x, n_x$ where n_x is a fresh value that serves two purposes (i) to provide the freshness guarantee for each runs in order to prevent from replay attack and (ii) to derive the secret key. Refinement $R_6, SIG_{x\,(m)} = SIG_x(m), ID_x$, where Id_x refers to public key certificate for x and the verification keys don't possess for others. The principles posses each other's public key certificate as an assumption considered before session establishment. Further, Refinement $R_7, SIG_x(m) = E_k(m), HMAC_{k'}(role, E_k(m))$ where k' and k identifies the protocol shared among initiator and responder. Just Fast Key formulation has used using this refinement. A keyed hashed includes encrypted signature.

Transformations at the other end classifies into three parts such as- Message Content Move T_1, Binding T_2 and Cookie T_3. A message T_1 moves from one state to another but any freshly generated data doesn't contain by the same. Transformation T_2 adds (in general) information from one part of a protocol to another in order so to bind the two parts in some meaningful way as:

$$\begin{array}{ll} I \to R: m & I \to R: m \\ R \to I: n, SIG_R(m) & \Rightarrow \quad R \to I: n, SIG_I(m, n) \\ R \to I: n, SIG_R(m) & I \to R: SIG_I(m, n) \end{array} \quad (3)$$

The Cookie T_3 transformation is a freshly generated data stores in small that makes a protocol resistant to DOS (blind) attack. When each time user logins to website the broser sends a cookies back to server for each user activity that can also be consideres on website previous activity.

For derivation of protocols an innovative approach has proposed, depicted in Figure 2, by Datta. We are presenting a broad aspect towards the protocol derivation using compositional logics:

Protocol P_1 obtaines from two symmetric component C_2 as sequential composition as:

$$I \to R: m; R \to I: SIG_R(m); I \to R: n; R \to I: SIG_I(n) \quad (4)$$

The m and n values assumes as fresh nonse and public key certificates for I and R poses each other's to verify the signatures.

Protocol P_2: Transformation T_1 applies on protocol P_1 to get this protocol:

$$I \to R: m; R \to I: n, SIG_R(m); R \to I: SIG_I(n) \quad (5)$$

This is a way to reduce the messages complexity length from 4 to 3.

Protocol P_3: This protocol achieves from protocol P_2 by T_2 binding operation:

$$I \to R: m; R \to I: n, SIG_R(m, n); R \to I: SIG_I(m, n) \quad (6)$$

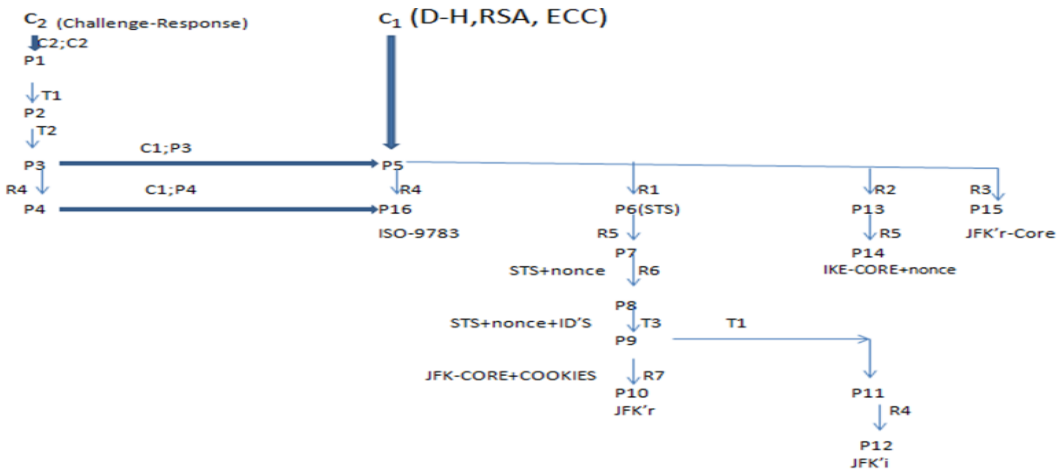


Figure 2. Protocol Derivation System (PDS) Approach

***Protocol P_4*:** This is one of the standard challenge-response protocol for the alternative derivation of ISO-9798-3 protocol, obtained by applying refinement R_4 over the protocol P_3.

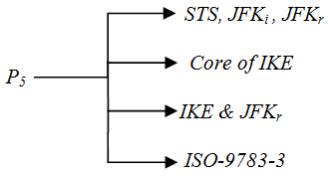
$$I \to R: m;\ R \to I: n, SIG_R(n, m, ID_I);\ R \to I: SIG_I(m, n, ID_R) \quad (7)$$

In this regard protocol P_3 has refined that includes inside the peer's identity signature, from man-in-middle attack, in case of wrong identities verification. Thus it provides the mutual authentication.

Protocol P_5: Component C_1 composes with protocol P_3, here C_1 is Diffie-Hellman component to get this protocol. ECC or RSA component also works according to needs is also applicable and also all remaining things are same.

$$I \to R: g^i;\ R \to I: g^r, SIG_R(g^r, g^i);\ R \to I: SIG_I(g^i, g^r) \quad (8)$$

If responder R is honest, initiator I completes its session with R, then a secret key g^{ir} share to them. Here a possible attack of man-in-the-middle attack is still possible. But, to overcome from situation four alternative paths for same as:



***Protocol P_6*:** This protocol obtaines from protocol P_5 by applying the refinement R_1. This is also known as STS protocol, it keeps all the properties of P_5 but in addition to it doesn't provide from identity protection from passive attackers, mutual authentication shared secret and man-in-middle attack.

$$I \to R: g^i;\ R \to I: g^r, E_K\left(SIG_R(g^r, g^i)\right);\ R \to I: E_K\left(SIG_I(g^i, g^r)\right) \quad (9)$$

However, man-in-middle attack is possible in this protocol which as proved Lowe here but he was not able to say for mutual authentication breaks the same or not.

Protocol P_7: On protocol P_6 refinement R_5 applies for this protocol:

$$I \to R: g^i, n_i;\ R \to I: g^r, n_r, E_K\left(SIG_R\left(g^r, n_r, g^i, n_i\right)\right); I \to R: E_K \quad (10)$$

The exponentials reuse across multiple sessions is computationally more efficient in this protocol is a motivation issues that makes it enable for the forward secrecy and it doesn't compromise its secrecy in the long run.

***Protocol* P_8**: Obtained by using refinements R_6 to protocol P_7, such as

$$I \to R: g^i, n_i;\ R \to I: g^r, n_r, E_K\left(SIG_R\left(g^i, n_i, g^r, n_r\right), ID_R\right);$$
$$I \to R: E_K\left(SIG_R\left(g^r, n_r, g^i, n_i\right), ID_I\right) \quad (11)$$

After the application of this refinement, the protocol assumption possessed a public key certificate that discharged from exchanging certificates alongside the signature identifiers, and by the other means that no new properties introduces.

***Protocol* P_9**: This protocol is to attained from the cookie transformation T_3 on protocol P_8:

$$\begin{cases} I \to R: g^i, n_i;\ R \to I: g^r, n_r, HMAC_{HK_R}\left(g^r, n_r, g^i, n_i\right); \\ I \to R: g^i, n_i, g^r, n_r, HMAC_{HK_R}\left(g^r, n_r, g^i, n_i\right), E_K\left(SIG_I\left(g^i, n_i, g^r, n_r\right), ID_I\right); R \to I: E_K\left(SIG_R\left(g^r, n_r, g^i, n_i\right), \right. \end{cases}$$
(12)

In addition to protocol properties P_8, this ensures the additional property that is resistant to the blind Denial-of-Service attacks.

At this juncture, the derived protocol provides the DoS protection, mutual authentication, key secrecy, computational efficiency and identity protection from initiator & responder, respectively.

Further, the Just Fast Key (JFK) for initiator (I) and responder (R) obtained from protocol P_9, with only differences they offeres in identity protection only.

***Protocol* P_{10}**: The JFK_R obtained from by applying refinement R_7 to protocol P_9. Instead of that, the protocol has added two more refinements.

$$\begin{cases} I \to R: g^i, n_i;\ R \to I: g^r, n_r, HMAC_{HK_R}\left(g^r, n_r, g^i, n_i\right); \\ I \to R: g^i, n_i, g^r, n_r, HMAC_{HK_R}\left(g^r, n_r, g^i, n_i\right), E_K\left(SIG_I\left(g^i, n_i, g^r, n_r\right), ID_I\right), HMAC_{K'}\left(I, E_K\left(SIG_I\left(g^i, n_i, g^r, \right.\right.\right. \\ R \to I: E_K\left(SIG_R\left(g^r, n_r, g^i, n_i\right), ID_R\right), HMAC_{K'}\left(R, E_K\left(SIG_R\left(g^r, n_r, g^i, n_i\right), ID_R\right)\right) \end{cases}$$
(13)

During computation the keys K and K' requires knowledge of g^{ir} that is a guarantees to initiate from the Man-in-the-middle attack and can't be computed the hashed encrypted signature.

***Protocol* P_{11}**: The JFK_I obtained from by applying transformation T_1 to protocol P_9. Instead of same, this protocol added modifications.

$$\begin{cases} I \to R: g^i, n_i;\ R \to I: g^r, n_r, ID_R, HMAC_{HK_R}\left(g^r, n_r, g^i, n_i\right); \\ I \to R: g^i, n_i, g^r, n_r, HMAC_{HK_R}\left(g^r, n_r, g^i, n_i\right), E_K\left(SIG_I\left(g^i, n_i, g^r, n_r\right), ID_I\right); R \to I: E_K\left(SIG_R\left(g^r, n_r, g^i, n_i\right), \right. \end{cases}$$
(14)

Here, the ID_R message component is shifted, to reason for applying transformation to include the peer's identity inside the signature. In this regards, I's signature posses the R's identity before to send the message in the protocol. This also retains all the properties contained by P_9 is different except for identity protection. But, the major drawback is the responder's identity protection.

Protocol P_{12}: The protocol P_{12} obtaines from the protocol P_{11} by applying refinement R_4. This is equivalent to JFK_i except for one additional signature added using the one more transformation in message and for other end the core security property ignored.

$$\begin{cases} \mathrm{I} \rightarrow \mathrm{R}: \mathrm{g^i}, \mathrm{n_i};\ \mathrm{R} \rightarrow \mathrm{I}: \mathrm{g^r}, \mathrm{n_r}, \mathrm{ID_R}, \mathrm{HMAC_{HK_R}}(\mathrm{g^r}, \mathrm{n_r}, \mathrm{g^i}, \mathrm{n_i}); \\ \mathrm{I} \rightarrow \mathrm{R}: \mathrm{g^i}, \mathrm{n_i}, \mathrm{g^r}, \mathrm{n_r}, \mathrm{HMAC_{HK_R}}(\mathrm{g^r}, \mathrm{n_r}, \mathrm{g^i}, \mathrm{n_i}), \mathrm{E_K}(\mathrm{SIG_I}(\mathrm{g^i}, \mathrm{n_i}, \mathrm{g^r}, \mathrm{n_r}, \mathrm{ID_R}), \mathrm{ID_I}); \mathrm{R} \rightarrow \mathrm{I}: \mathrm{E_K}\left(\mathrm{SIG_R}(\mathrm{g^r}, \mathrm{n_r}, \mathrm{g^i}, \mathrm{n_i}, \mathrm{ID}\right. \end{cases}$$

(15)

The peer's identities refinement adds of ID_I and ID_R inside the signatures, respectively. This prevents the attacks, and retains all the properties of protocol P_{11}.

Protocol P_{13}: The Internet Key Exchange (IKE) is one of the protocol that have obtained from applying refinement R_2 to protocol P_5. This has described as the core for IKE as

$$I \rightarrow R: g^i;\ R \rightarrow I: g^r, SIG_R\left(HMAC_K(g^r, g^i, ID_R)\right); I \rightarrow R: SIG_I\left(HMAC_K(g^i, g^r, ID_I)\right) \quad (16)$$

Each principal used in the exponentials signs a keyed hash and their own identities. The adversary can't attack on the used hashed key from the secret g^{ir} which is only known to I and R. So, this provides both to mutual authentication and a shared secret between them.

Protocol P_{14}: This protocol derivation achieved using the refinement R_5 applied to protocol P_{13}. This sensibly parallels the steps for JFK_r and JFK_i where exponential nonces exchanged.

$$I \rightarrow R: g^i, n_i;\ R \rightarrow I: g^r, n_r, SIG_R\left(HMAC_K(g^r, n_r, g^i, n_i, ID_R)\right);$$
$$I \rightarrow R: SIG_I\left(HMAC_K(g^i, n_i, g^r, n_r, ID_I)\right) \quad (17)$$

The purpose is to allow and reuse the exponential in a more efficient protocol for multiple sessions. Although, it contains one of the tradeoff during the processing, in the loss of perfect forward secrecy.

Protocol P_{15}: One of alternative path for protocol P_5 that consists of the core for JFK_r and $JFK - SIGMA$. This protocol has obtained using the refinement R_3 to protocol P_5.

$$I \rightarrow R: g^i;\ R \rightarrow I: g^r, SIG_R(g^r, g^i), HMAC_K(g^r, g^i, ID_R);$$
$$I \rightarrow R: SIG_I(g^i, g^r), HMAC_K(g^i, g^r, ID_I) \quad (18)$$

This is a very similar protocol like P_{13}, that also possesses the same properties of shared secret and mutual authentication. One of differences observation is in signing the keyed hash and the principals to send the hash separately. So, for adversary can't launch the MAN-IN-MIDDLE ATTACK attack because the computation of the hash requires only known to I and R.

Protocol P_{16}: This protocol obtaines from protocol P_5 by using refinement R4, it also known as ISO-9783-3 protocol:

$$I \rightarrow R: g^i; R \rightarrow I: g^r, SIG_R(g^r, g^i, ID_I); I \rightarrow R: SIG_I(g^i, g^r, ID_R) \quad (19)$$

This protocol set-up a secret mutual authentication scheme and refers to man-in-middle attack is not possible, because of its intended identity recipient's signature, attacker doesn't forward identity either to I or R.

Now, we are applying the recent protocol of Multilayer Consensus ECC-Based Password Authenticated Key Exchange Protocol (MCEPAK) as a key component, the next section elaborates for the same and is also one of the motivating issue.

4. Related Work AnD Background

Multilayer Consensus ECC Based Password Authenticated Key Exchange Protocol (MCEPAK) is a key for our work and its related idea has been presented in [20]. Password authenticated key exchange protocol (PAKE) is an elementary protocol derivation based on two-step and it is also known as Simple Authenticated Key Exchange Protocol (SAKA), presented in [21], shown in Figure 3. Further, X. Ding *et. al.*, presented a three step PAKE to resist password compromise impersonation, ephemeral key compromise, forward secrecy, and dictionary attack. The IEEE 1063.2 standard was released in 2009. This standard scheme specifies low-grades secrets as a stronger security basis for securing transactions and to show a proficiently utilizing passwords [22]. The basic idea is presenting here, a group generaror G is available, where each party randomly select its secret keys (as a number) and multiplies the same with G, which shares using the ECC as depicted in $X.1035$ standard that is resistant to the password guessing attack.

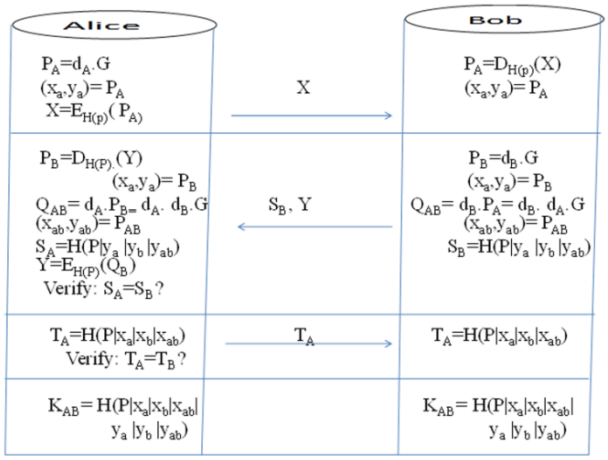


Figure 3. Working of ECC-based PAKE (EPAK) Protocol in between Alice & Bob

Step I: Let us assume that Alice is a initiator that elect to choose a secret random key d_A and multiply with group generator G to obtain the public key P_A and represented the same to Elliptic point (x_a, y_a). It computes hash $H_{(p)}$ to obtain a symmetric key with which encrypts P_A as X and send Bob (20):

$$P_A = d_A.G; (x_a, y_a) = P_A; X = E_{H(P)}(P) \quad (20)$$

Upon receiving a packet X, Bob decrypts the same and represents in elliptic point (x_a, y_a) (21):

$$P_A = D_{H(p)}(X); (x_a, y_a) = P_A \quad (21)$$

Step II: Bob picks a secret random key d_B as private key and to obtain public key P_B it multiplies to group generator G, also appropriate Elliptic point in (22):

$$P_B = d_B.G; (x_a, y_a) = P_B \quad (22)$$

Again, multiplies private key to Alice public key to obtain a shared key Q_{AB} and finds its appropriate EC points $(x_{ab}, y_{ab}) = P_{AB}$ then computes S_B having $Q_A, Q_B \; and \; Q_{AB}$ and finally uses $H_{(P)}$ to encrypt:

$$Q_{AB} = d_B.P_A = d_B.d_A.G; (x_{ab}, y_{ab}) = P_{AB}; S_B = H(P \mid y_a|y_b|y_{ab}); Y = E_{H(P)}(P_B) \quad (23)$$

To decrypt Y, Alice uses $H_{(P)}$ and obtains Q_B and also the converted elliptic point (x_a, y_a) aligned to Q_B. Again, the Alice private key multiplies with public key sent from Bob Q_B and shares a common shared key Q_{AB} followed to points (x_{ab}, y_{ab}). Finally she computes S_A for verification of having the values of $Q_A, Q_B \; and \; Q_{AB}$. If the verification holds, she can sure that Bob has the required values as (24):

$$P_B = D_{H(P)}(Y); Q_{AB} = d_A.P_B = d_A.d_B.G; S_A = H(P|y_a|y_b|y_{ab}) \quad (24)$$

Step III: Alice needs to make assure to Bob that she has values as well. So, need to performs T_A out of Q_A, Q_B and Q_{AB} and send it to Bob as (25):

$$T_A = (H(P|x_a|x_b|x_{ab}) \quad (25)$$

The other side Bob calculates T_B and compares with T_A. If the verification holds Bob assures Alice also is the required values as well (26):

$$T_B = H(P|x_a|x_b|x_{ab}) \quad (26)$$

Step IV: So far, both parties have the required parameters and verifies to each other. Finally, they perform to calculate the secret shared key as (27):

$$K_{AB} = H(P|x_a|x_b|x_{ab}|y_a|y_b|y_{ab}) \quad (27)$$

Further, here MCEPAKE protocol for key exchange consideres. A key agreement for mutual authentication among (an initiator) appliance network A_N, Home Area Network H_c, Building Area Network B_c, Neighbor Area Network N_c and Central Controller C_c has considered. Where each intermediate controller resulted in the form of individual key in them are correctly working for all. In this we consideres the same approach which takes the advantages of ECC for key generation. The approach provides a high level of security with the smaller key size, exposes in Figure 4. If required, the presented protocol can be extendes to a larger layers of security, it can also be implement by the adaption of the current $X.1035$ standard and applies for ECC.

Using ECC approach, the first iteration between A_N and H_c for PAKE is based. Further, the same philosophy applies for second, third and fourth consequent layers. These have been available for multilayer consensus on password authenticated key exchange protocol for ECC.

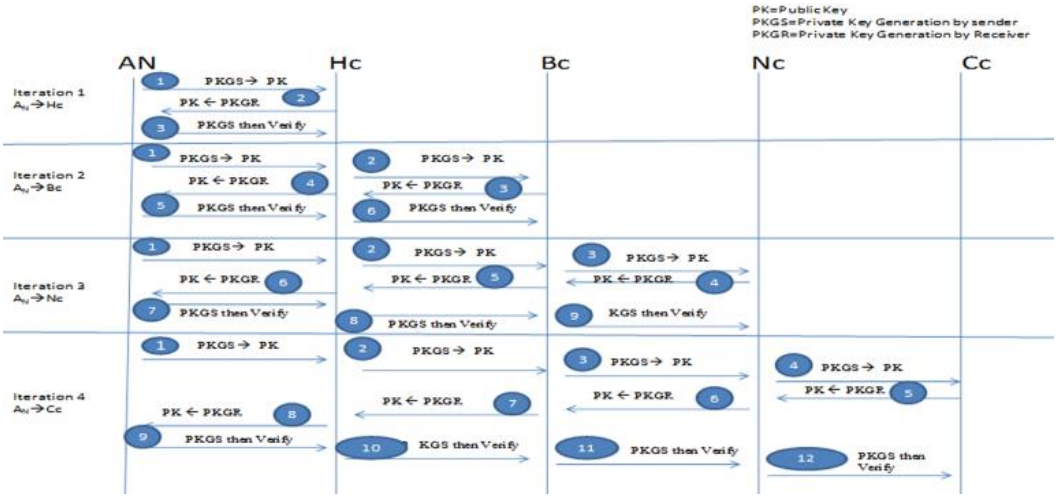


Figure 4. Multilayer Consensus Key-Generation Approach

Since the MCEPAK proposed protocol [20] is established on ECC, and X.1035 that contains similar benefits like the Diffie-Hellman procedure. In the proposed work, the security and different attacks have analyzed and modeled on the same, where an adversary (internal or external) is capable of re-scheduling, re-playing, re-ordering, re-routing, deleting and recording the messages considered. We have done the formal verification of the proposed protocol underneath under the similar adversary conditions.

5. Formal Validation Using Span and Avispa Tool

AVISPA [23] is one of the automatic verification and validation tool that used in the cryptography. It has insidious for Internet security applications and its protocols. It offers a significant expressive formal language for specifying protocols with their safety measures that has modularized into different four back-ends under the perimeter, the structure shown in Figure 5 as:

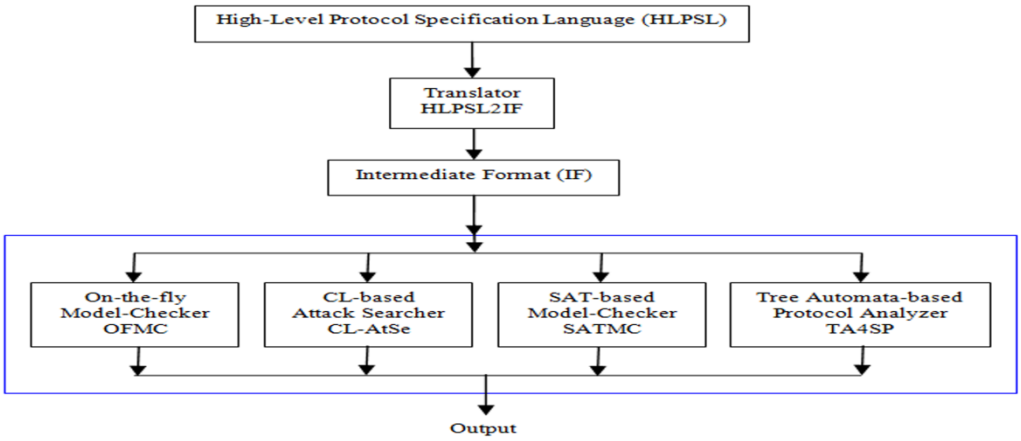


Figure 5. AVISPA Structure

Its accomplishment is based on the automatic analysis techniques. The High Level Protocol Specification Language (HLPSL) describes to formally validate the security protocols and as well as it specifies the intended security properties. The HLPSL specification first translated into Intermediate Format (IF) through translator HLPSL2IF. Where the IF is a lower-level language and, that, is directly interpreted for back-ends tool. The IF objective has formulated for developers with the implication to use as their input

language analysis. This happens automatically and is transparent to the user [24]. Now, the IF specification analyzed at the back-ends for the satisfied or violated security goals. The AVISPA Tool comprises four back-ends such as: On-the-fly Model Checker (OFMC) [25], Constraint Logic-based Attack Searcher (CL-AtSe) [26], SAT-based Model Checker (SATMC) [27]-[28], and Tree-Automata Based Protocol Analyzer (TA4SP) [29]. The definition of OFMC says it is a useful debugging tool for protocol specification that allows agents to execute all the required steps for honest run of the protocol and specific for manual check if needs. CL-ATSE is set of constraints, uses to find attacks on protocols where translation and checking are fully automatic that are internally performed by the same i.e. no external tool uses. Its back-end traces uses a slightly different format for some aspects of attack than OFMC does. For example, it writes an interpretation in the intermediate format (IF) as tests or actions in the attack trace. SATMC's is used to check the executability that includes functionality to confirm the executability of a HLPSL specification. SATMC is particularly strict about the proper use of types in HLPSL specifications; this feature can thus be very useful for finding errors relating to typing that may lead to non-executability of a protocol specification. The TA4SP proves secrecy properties with an unbounded number of sessions. From the practical point of view, this works completely automatic and supported by two (2) tools such as Timbuk and its extensional part. The analysis of four back-ends are harmonized to each others in a sense for some common back-ends procedure, but these are not equivalent that should return different results. The proposed MCEPAK protocol running on the tool, shown in Table 1, at back ends of OFMC and CL-AtSe, with safety measures.

Table 1. OFMC and Cl-AtSe Back end results on AVISPA

A@ubuntu:~/avispa-1.1$avispa BasicMainHlPsl.hlpsl --ofmc	**A@ubuntu:~/avispa-1.1$ avispa BasicMainHlPsl.hlpsl --cl-atse**
% OFMC % Version of 2006/02/13 SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/A/avispa-1.1/testsuite/results/BasicMainHlPsl.if GOAL as_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00s searchTime: 0.05s visitedNodes: 6 nodes depth: 2 plies	SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/A/avispa-1.1/testsuite/results/BasicMainHlPsl.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 1 states Reachable : 0 states Translation: 0.02 seconds Computation: 0.00 seconds

An impressive SPAN tool comes with simple editing protocol specifications of web graphical interfaces of AVISPA, and in addition to this it contains honest agents for protocol simulation, intruder simulation for honest agents and an attack simulation. Attack simulation in this, like the same layout as intruder simulation, but attacks are automatically built using OFMC/CL-AtSe facilities. The security protocol analysis is the major idea possible through the two specification such as High Level Protocol Specification Language (HLPSL) and CAS+. HLPSL is a used language for specifying the cryptographic protocols for AVISPA toolset and CAS+ is a light evolution of CASRUL language. The Figure 6, says the ECC operation in CAS language and shows the principles of sender pattern as the tool dictates like the same.

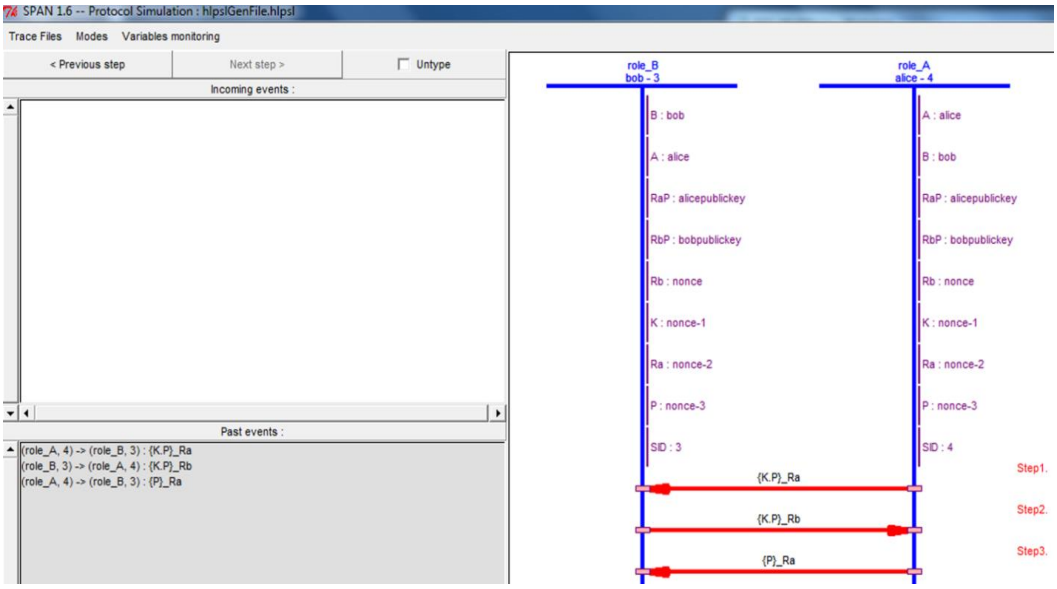


Figure 6. ECC Protocol Verification in view of Sender Pattern Principal

This specification translates a CAS+ language from Alice-to-Bob for fast and simple specification of security protocols; interactively building a Message Sequence Chart (MSC) [30]-[31] of protocol execution; MSC automatically build attacks on either of HLPSL and CAS+ specifications; and intruder mode interactively build specific attacks on specifications.

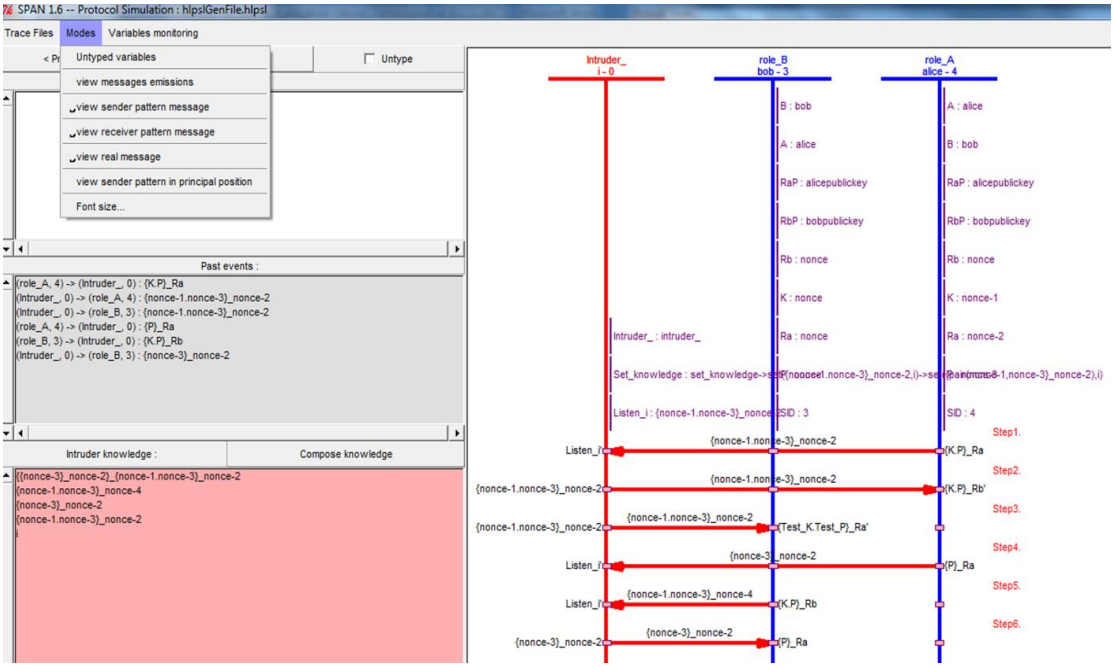


Figure 7. Intruder Simulation on ECC in between Sender and Receiver

Figure 7 has shown ECC-Intruder Simulation with sender-receiver pattern and real messages transformation observation.

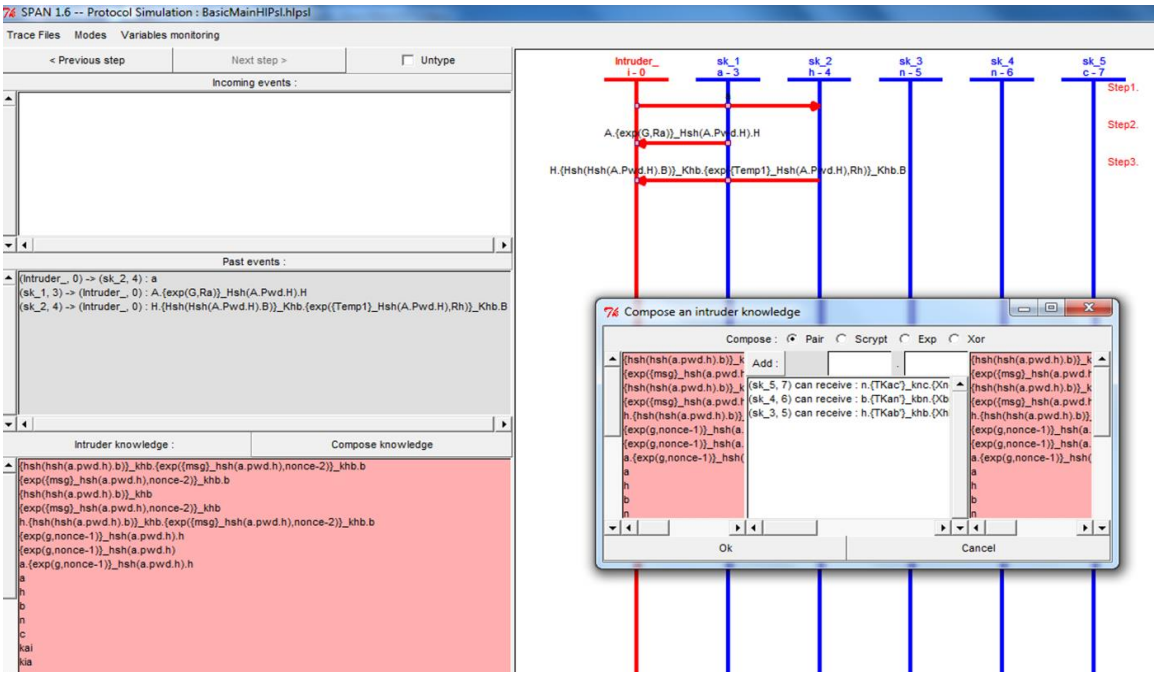


Figure 8. Intruder Simulation on Multilayers Consensus Protocol

In Figure 8, we have shown the intruder formal verification for multilayer consensus. The MCEPAKE proposed scheme is a enhancement for multilayer security among the layers of networks. Its percent improvement encryption and decryption time between layers presents in below Table 2.

Table 2. Execution Time: Encryption and Decryption

$A_N \leftrightarrow H_C$	$A_N \leftrightarrow B_C$	$A_N \leftrightarrow N_C$	$A_N \leftrightarrow C_C$
$2\times t_0$	$4\times t_0$	$6\times t_0$	$8\times t_0$
$2\times t_0$	$2\times t_0$	$2\times t_0$	$2\times t_0$
0%	50%	66.69%	75%

Using the ECC algorithm for encryption and decryption of desired message m with the shared secret key K uses in MCEPAKE protocol to form the ciphertext $C_m = \{KG, m + K.P_B\}$, where P_B is the responder public key used by the initiator. Now, to decrypt the ciphertext C_m to get the original plain texts message $m = \{m + K.P_B - n_B(K.G)\}$ where n_B is the private key of responder [32], is using during the whole procedure.

In the next section, our proposed Signcryption technique has presented in relation to the basics of this technique. This is one the efficient and moderninized technique that serves the two purposes such as digital signature and encryption of transmitted message with reduced computation and communication costs.

6. Signcryption

In 1997 Zheng [33] was first proposed the Signcryption primitive of cryptography. It logically combines digital signature and encryption scheme in a single step on less computational and communication cost. Using the Signcryption, he proposed 58% of less computational cost and 70% less communication cost only needs when in general it compares with the individually signature–then-encryption schemes. The parameters used in schemes contains the respective sizes that decides it cost such as $|p| = 512$ bits, $|q| = 144$ bits and $|hash(.)| = |KH(.)| \cong |q|/2$. Here we presented a Table 3 that shows the above enrichments in relation to basic signature-then-encryption on Schnoor

Signature plus ElGamal Encryption versus Zheng signcryption. Except the exponential (EXP) function, the other functions used in the explanation are equalized to each other so its cost considered to be negligible. The computational cost represents a reduction in $\{(5.17-2.17)/(5.17)\}=58\%$. Whereas this table concludes the saving in communicational cost $\{(|\text{hash}(.)|+|\text{p}|+|\text{q}|)-(|\text{KH}(.)|+|\text{p}|)/(|\text{hash}(.)|+|\text{p}|+|\text{q}|)\}=70\%$.

Table 3. Cost of Signature-then-Encryption versus Cost of Signcryption

Schemes	Computational Cost	Communicational Cost
Signature-then-encryption based on "Schnoor Signature + ElGamal Encryption"	EXP=3, MUL=1,DIV=1,HASH=1,ENC=1 { EXP=2.17, MUL=0,DIV=1,HASH=2,ENC=1} Total Modular Reduction=5.17	$\|hash(.)\|+\|p\|+\|q\|$
Signcryption	EXP=1, MUL=1,DIV=1,HASH=1,ENC=1 { EXP=1.17, MUL=0,DIV=1,HASH=2,ENC=1} Total Modular Reduction=2.17	$\|KH(.)\|+\|p\|$

This is a huge saving for applications in computation and communication cost in the form of secure & authenticated message delivery. There are other applications that bring into notices such as authenticated electronic secure transactions, non-repudiated key transportation, inclusive video conferencing through secure and authenticated multicast services, unforgeable messages fast & compact services.

Since throughout the years, there are many variations of this scheme have been proposed by having its own problems and limitations, with offering optimized computational costs and different levels of security. Baek in 2002 given the formal proofs of Signcryption in [34]. The real life application of Signcryption is based on "*killing two birds by one stone*". Confidentiality is achieving through encryption, whereas authentication is providing the integrity for this scheme. Private key authentication and public key digital signatures are authentication schemes are playing an important scheme.

In general, the objective of Signcryption states that it should satisfy this condition: Cost (Signature and Encryption) << Cost (Signature) + Cost (Encryption) [33]. Further, these are interpreted in a number of ways:

• A combination of digital signatures and encryption scheme, signcryption should be more efficient (computationally).

• A naive combination of digital signatures and ciphertext encryption, signcryption should produce shorter cipher text.

• A naive combination of digital signatures and public-key encryption, signcryption should endow with better safety measures and/or bigger functionality compared.

The scheme of signcryption scheme works in five phases such as: Setup phase, Sender Key Generation phase, Responder Key Generation, Signcrypt phase, and in Unsigncrypt phase.

Phase 1: Setup Phase

The setup factor defined for the same is based on the security and common key generation parameters. The overall parameters public for all as a summary contains like: p is a large prime; q is a prime factor of $p-1$; g is an integer with order q modulo p in $[1,\dots,p-1]$; KH is key hashed one way function of hash(k,m); and (E,D) are used for encryption E and decryption D

Phase 2: Sender Key Generated Phase

Alice has the pair of keys (X_a, Y_a), where X_a Alice private key randomly chosen from $[1, \ldots, q-1]$; Y_a is a generated public key for Alice to modulo prime p; Alice now ready to send a message to Bob

Phase 3: Responder Key Generation Phase

Bob keeps a pair of keys (X_b, Y_b), as private key X_b randomly selected from $[1, .., q-1]$ and his public key Y_b generated on the prime modulo p. Bob is now ready to send a message to Alice

Phase 4: Signcrypt Phase

The initiator and responder accomplishes the following operations in order to signcrypt a message:

The key k splits in k_1 and k_2 of equal length parts; Calculate $= hash(k_2, m)$; Calculate $s = x/(r + X_a)$ mod q; Calculate $c = E_{k_1}(m)$ = the encryption of the message m with the key k_1; Alice sends to Bob the values (r, s, c)

Phase 5: Unsigncrypt Phase

Finally, in order to Unsigncrypt the sighpcrypted message, Responder accomplishes the following operations:

Calculate $k\ using\ r, s, g, p, Y_a and\ X_b$; $k = hash(Y_a * g^r)^{s*X_b} \bmod p$; now again Split k in k_1 and k_2 for the verification of original message in the form of appropriate lengths; the message m evaluates by performing decryption $m = Dk_1(c)$; A valid message m accepted only if $KHk_2(m) = r$ satisfy.

Table 4. Comparison of Different Algorithm Schemes based Operations

Schemes	Participant	ECPM	ECPA	DIV	MUL	ADD	HASH
Zheng	Sender	1	-	1	1	1	2
	Receiver	2	1	-	2	-	2
Hwang	Sender	2	-	-	1	1	1
	Receiver	3	1	1	-	-	1
Zhou	Sender	2	2	1	2	1	3
	Receiver	4	4	-	1	1	3
Basu	Sender	2	-	-	2	1	1
	Receiver	3	1	1	1	1	1
Proposed Scheme	Sender	1	1	2	-	1	1
	Receiver	1	-	2	2	-	1

Signcryption contains the various unique features such as: is much smaller overhead requires than the conventional sign-then-encrypt schemes, security against unforgeabilty, unsigncryptabilty to verify message. The Table 4, is our proposed scheme shows the improvement over Basu et al. [35] and its related proposed schemes on elliptic curve point multiplication (ECPM), Multiplication MUL and addition ADD.

The correctness definition of the scheme is secure, if it satisfies the following conditions:

(i) Unforgeabilty: For an adaptive attacker, it is computationally infeasible for the dishonest Bob and then to allow querying for Alice signcryption to masquerade in creating authentic text messages.

(ii) Non-repudiation: For a third party, it is computationally feasible to settle the dispute between the two events; Alice denies the fact that she is the originator of a signcrypted text with Bob as its recipients.

(iii) Confidentiality: For an attacker it is computationally infeasible to gain the partial information from the signcrypted text. The other party involved may be anyone other than Alice/Bob.

Further, the scheme has generalized into the forms of requirements specifications. It is not only necessary for all messages requires integrity and confidentially. Where some messages requires sign only, while others need to be encrypted. Later on the two cases may provides one of the specific parties to them, despite the fact that conventional signcryption requires both of them. As a result the applications must implement into the three individual primitives that include signature, encryption, and signcryption. This scheme has generalized that provides the dual functions with more practicability and flexibility, when simultaneously requires authenticity and confidentiality. Also, it is endowed with solitary signature or encryption function when authenticity/confidentiality requires without any additional computation and amendments [36].

In the recently scenario, there are many applications are in light due to its various ability such as: its decreased computation cost, reduced bandwidth, easily applicable to tiny digital phone, wireless transport layer security handshake protocol, and the ability to connect to the internet. Unforgeable key establishment is the second major application over ATM networks.

7. Conclusion

This manuscript contains a secure composition approach that adds and/or makes a way for secure computing techniques. These approaches are widely contributing the significant importance in the cryptographic applications. Instead of the same our focus is relative advantages over the signcrypted multilayer consensus based approaches for secure composition. It is showing in information security, the proposed approaches makes a scientifically strong security mechanisms in applied cryptography. Our proposed approach has considered the protocol derivational system and protocol compositional logic approach. The abstract idea presents to derive the use of basic components in the formation of Diffie-Hellman, and applicability for secure composition that can also apply for ECC with its reduced relative cost. In addition to same, security concerns without any compromise, it offers a faster computation and requires less memory. Then after, using the signcryption primitive has applied on multilayer consensus ECC based, password-authenticated key exchange protocol approach that drastically reducing both to computational and communicational cost. Where, new paradigm of signcryption applies for cost effectiveness, high performance, favourable for short-memory devices applications and many more are the possible advantages on the proposed approaches. Instead of that the protocol has formally validated on AVISPA and SPAN tools.

References

[1] A. Datta, "Security Analysis of Network Protocol: Compositional Reasoning and Complexity-Theoretic Foundations", PhD Thesis, Department of Computer Science Stanford University, Online: http://seclab.stanford.edu/pcl/papers/datta-thesis.pdf, **(2005)**.

[2] A. Datta, A. Derek, J.C Mitchell and A. Roy, "Protocol Composition Logic", Electronics Notes in Theoretical Computer Science (ENTCS), Elsevier, , DOI: 10.1016/j.entcs.2007.02.012, vol. 172, **(2007)** April, pp. 311-358.

[3] A. Datta, A. Derek, J.C Mitchell and D. Pavlovic, "A Derivation System for Security Protocols and its Logical Formalization", in Proceeding of 16th IEEE Computer Security Foundations Workshop, **(2003)**, pp. 109-125.

[4] M. Borrows, M. Abadi and R. Needham, "A logic of authentication", ACM Transaction on Computer Systems, vol. 8, **(1990)**, pp. 18-36.

[5] W. Diffie and M.E. Hellman, "New Directions in Cryptography", IEEE Transaction on Information Theory, DOI: 10.1109/TIT.1976.1055638, vol. 22, **(1976)** Nov., pp. 644-654.

[6] K. Jarvinen and J. Skytta, "Parallelization of High-Speed processor for Elliptic Curve Cryptography", IEEE Transaction on VLSI, vol. 16, **(2008)** Sep., pp. 1162-1175.

[7] N. Koblitz, "Elliptic Curve Cryptosystems", Math Computation, vol. 48, **(1987)**, pp. 203-209.

[8] V.S. Miller, "Use of Elliptic Curves in Cryptography", Advances in Cryptology, DOI: 10.1007/3-540-39799-X_31, **(1986)**, pp. 417-426.
[9] I. F. Blake, V. K. Murty and G. Xu, "A note on window τ-adic NAF algorithm", in Information Processing letters, DOI: 10.1016/j.ipl.2005.05.013, vol. 95, **(2005)** Sep., pp. 496-502.
[10] D. R. Hankerson, A Menezes and S. Vanstone, "Guide to Elliptic Curve Cryptography", Springer, **(2004)**, DOI: 10.1007/b97644.
[11] S. Arno and F. S. Wheeler, "Signed Digit Representations of Minimal Hamming Weight", IEEE Transaction on Computers, DOI: 10.1109/12.238495, vol. 2, no. 8, **(1993)** Aug., pp. 1007-1010.
[12] P. Longa and A. Miri, "Fast and Flexible Elliptic Curve Point Arithmetic over Prime fields", IEEE Transaction on Computers, vol. 57, **(2008)** Mar., pp. 289–302.
[13] E. Amini, Z. Jeddi and M. Bayoumi, "A High-Throughput ECC Architecture", in 19th IEEE Int'l Conf. on Electronic, Circuits and Systems, DOI: 10.1007/978-3-540-89255-7_20, **(2012)** Dec., pp. 901-904.
[14] T. Izu and T. Takagi, "Fast Elliptic Curve Multiplications with SIMD operations", Information and Communication Security, Lecture Notes in Computer Science, DOI: 10.1.1.97.5074, vol. 2513, **(2002)** Dec., pp. 217-230.
[15] W. Fischer, C. Giraud, E. W. Knudsen and J. -P. Seifert, "Parallel Scalar Multiplication on General Elliptic Curves over F(p) hedged against Non-Differential Side-Channel Attacks", IACR (2002/007), Cryptology ePrint Archive. [Online] http://eprint.iacr.org/2002/007.
[16] P. K. Mishra, "Pipelined Computation of Scalar Multiplication in Elliptic Curve Cryptosystems (Extended Version)", IEEE Transaction on Computers, DOI: 10.1109/TC.2006.129, vol. 55, no. 8, **(2006)** Aug., pp. 1000–1010.
[17] Introduction to NISTIR 7628 guidelines for smart grid cyber security. National Institute of Standards and Technology (NIST), 2010. [Online]. Available: http://www.nist.gov/smartgrid/upload/nistir-7628_total.pdf.
[18] A. Datta, A. Derek, J.C Mitchell and A. Roy, "Protocol Composition Logic", in ELESEVIER, Electronics notes in Theoretical Computer Science, vol. 172, **(2007)**, pp. 311-358.
[19] Z. You, J. Tao and X. Li, "Extension and Application of Protocol Composition Logic", in Proceeding ICCET, IEEE, DOI: 10.11.9/ICCET.2010.5485720, vol. 4, **(2010)**, pp. V4-77-V4-81.
[20] H. Nicanfar and V.C.M Leung, "Multilayer Consensus ECC-Based Password Authenticated Key-Exchange (MCEPAK) Protocol for Smart Grid System", IEEE Transactions on Smart Grid, **(2012)**, pp. 1-12.
[21] D. H Seo and P. Sweeney, "Simple authenticated key agreement algorithm", *Electronic Letters*, Institution of Engineering and technology, DOI: 10.1049/el:19990724, vol. 35, **(1999)**, pp. 1073–1074.
[22] D. Xiao-fei, M. Chuan-gui and C. Qing-feng, "Password authenticated key exchange protocol with stronger security", in 1st Int'l Workshop (Wuhan, Hubei, China) Education Technology and Computer Science (ETCS'09), IEEE, DOI: 10.1109/ETCS.2009.411, vol. 2, **(2009)**, pp. 678-681.
[23] AVISPA-Automated Validation of Internet Security Protocols [online] Available: http://www.avispa-project.org.
[24] V. Sperschneider and G. Antoniou, "Logic: A Foundation for Computer Science", 1st ed., Addison-Wesley Longman, USA, **(1991)**.
[25] D. Basin, S. Modersheim and L. Vigano, "OFMC: A Symbolic Model-Checker for Security Protocols", International Journal of Information Security, DOI: 10.1007/s10207-004-0055-7, vol. 4, **(2005)**, pp. 181-208.
[26] M. Turuani, "The CL-Atse Protocol Analyzer", Lecture Notes in Computer Science, F. Pfenning, Eds. in RTE, DOI: 10.1007/11805618_21, vol. 4098, **(2006)**, pp. 277-286.
[27] A. Armando and L. Compagna, "Automatic SAT-Compilation of Protocol Insecurity Problems via Reduction to Planning", in Proceedings of FORTE 2002, Lecture Notes in Computer Science, DOI:10.1007/3-540-36135-9_14, vol. 2529, **(2002)**, pp. 210–225.
[28] A. Armando and L. Compagna, "Abstraction-driven SAT-based Analysis of Security Protocols", in Proceedings of TAST, Lecture Notes in Computer Science, DOI:10.1007/978-3-540-24605-3_20, vol. 2919, **(2004)**, pp. 257-271.
[29] Y. Boichut, N. Kosmatov and L. Vigneron, "Validation of Prouve Protocols using the Automatic Tool TA4SP", in 3rd Taiwanese-French Conf. on Information Technology, **(2006)**, pp. 467-480.
[30] D. Harel and P.S Thiagarajan, "Message Sequence Charts", UML for Real: Design of Embedded Real-time Systems, **(2003)**.
[31] D. Dolev and A. Yao, "On security of Public key protocols", IEEE Transactions on Information Theory, DOI: 10.1109/TIT.1983.1056650, vol. 29, **(1983)**.
[32] W. Stallings, "Cryptography and Network Security, Principles and Practices", 3rd Ed., Pearson Education, **(2004)**.
[33] Y. Zheng, "Digital signcryption or how to achieve Cost (Signature & Encryption) << Cost (Signature) + Cost (Encryption)", in Advances in Cryptology-CRYPTO'97, Lecture Notes in Computer Science, Springer-Verlag, vol. 1294, **(1997)**, pp. 165-179.
[34] J. Baek, R. Steinfeld and Y. Zheng, "Formal Proofs for the Security of Signcryption", in Public Key Cryptography (PKC 2002), Lecture Notes in Computer Science, Springer-Verlag, vol. 2274, **(2002)**, pp. 80-98.

[35] A Basu, I Sengupta and J.K Sing, "Formal Security Verification of Secured ECC Based Signcryption Scheme", Advances in Intelligent Systems and Computing, Springer Berlin, [Digests proceedings of the 2nd Int'l Conf. on Computer Science, Engineering & Applications, (ICCSEA), 2012]. DOI: 10.1007/978-3-642-30111-7_68, vol. 167, **(2012)**, pp. 713-725.
[36] Y. Han and X. Yang, "ECGSC: Elliptic Curve based Generalized Signcryption Scheme", Journal of Cryptology, **(2006)**. [Available: https://eprint.iacr.org/2006.pdf]

Authors

Gautam Kumar, received his Bachelor of Engineering (B.E) in Computer Science & Engineering from Rajiv Gandhi Proudyogiki Vishwavidyalaya (RGPV), INDIA in 2005, and his M.Tech in Computer Science & Engineering from Rajasthan Technical University, INDIA in 2012. Currently he is pursuing his PhD from Jaypee University of Information Technology, Waknaghat-173234, INDIA. His research interests are in the field of Cryptography, Network Security, Computer Networks and Algorithms. He is also having a teaching experience of 7.5 years for undergraduate/postgraduate students. He has published his research in journal and conferences of repute.

Hemraj Saini, is currently working as Assistant Professor (Senior Grade) in the Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat-173234 INDIA. He has received his PhD from Utkal University, Bhubaneswar, INDIA in 2012, M.Tech. from Punjabi University, Patiala, INDIA and B.Tech. from NIT Hamirpur, INDIA in 2005 and 1999, respectively. He is having more than 16 years of teaching and R&D experience. He has published around 100 research papers in journals and conferences of international repute. He has organized National and International conferences sponsored by agencies like IEEE, CSI, AICTE, CSIR, DST *etc*. He is the member of different professional technical and scientific associations such as IEEE (Mem. No. 92738007), ACM (Mem. No. 5156611), IAENG (Mem. No. 133186), *etc*. Presently he is providing his services in various modes like, reviewer for different reputed journals and conferences and also the Member of Editorial boards and Technical Program Committees.