International Journal of Advanced Engineering Research and Science (IJAERS) *Special Issue (NCCSIGMA-16)*
https://dx.doi.org/10.22161/ijaers/si.16 *ISSN: 2349-6495(P) | 2456-1908(O)*

Simple Secure Electronic Transaction (SSET) Protocol

Neha Kaliya[1], Dr. Muzzammil Hussain[2]

[1]Department of Computer Science & Engineering, Central University Of Rajasthan, Ajmer, India
Email: 2015mtcse010@curaj.ac.in
[2]Department of Computer Science & Engineering, Central University Of Rajasthan, Ajmer, India
Email: mhussain@curaj.ac.in

Abstract*—In the past years, a large number of proposals have turned up to execute electronic payments over internet securely. Electronic- commerce payments need to be more secure. Among all these proposals, SSL/TLS and SET are being installed world-widely for online credit card payments to be done securely. SSL protocol, due to its optional client authentication phase does not eliminate non-repudiation and thus merchant can also store delicate information of cardholder. SET ensures payment integrity, confidentiality and authentication of merchants and cardholders but at the same time, it is inefficient due to its complexity and overheads. Also, due to distribution of digital certificates and rules for client software installation, it is difficult to avoid and manage non-repudiation. Based on our study of SSL/TLS and SET, we proposed an upgraded version of SET protocol. It uses a highly secure session key sharing mechanism to ensure at most confidentiality, at the same time it ensures authenticity of the entities, integrity and also avoids non-repudiation. The proposed protocol make sure of enhanced security at a lower computational and storage costs.*

Keywords*—Confidentiality, E-Commerce, Integrity; Secure Socket Layer (SSL), Transport Layer security (TLS).

I. INTRODUCTION

E-Commerce (Electronic Commerce) refers to online business which does not include any physical exchange of goods and services rather all ordering, payments and delivery are done via internet [10].

It provides convenience, time-savings and many more benefits to the consumers as well as merchant and also increases efficiency. These E—commerce transactions are classified in various domains as: Business-to-Consumer (B2C), Business-to-business (B2B), Consumer-to-consumer (C2C), Business-to-Government (B2G) and mobile commerce (m-commerce). In this paper, B2C transactions and their security mechanisms are discussed. [10]

In B2C transactions, the generally used mechanism for online payment is credit card mode of payment. Hence, the credit card ID's are highly vulnerable and thus, security can be compromised [10].

We studied the requirements of E-commerce transactions security, various e-commerce protocols, their design issues, limitations and implementations and proposed a new efficient protocol to ensure transactions more securely. The main objective of our protocol is to provide a secure mechanism to transfer session key used during communication as well as to provide authentication and integrity. Section 2 gives overview of SET, Section 3 includes SET overheads, Section 4 describes the proposed protocol, Section 5 includes theoretical evaluation of secure mechanisms used and conclusion closes the paper in Section 6.

II. LITERATURE REVIEW

The Secure Electronic Transactions (SET) [4][5][6] is a protocol which has the ability to stand as a important factor in the security of e-commerce transactions. It was manufactured by Visa and MasterCard, with IBM and other computer vendors.

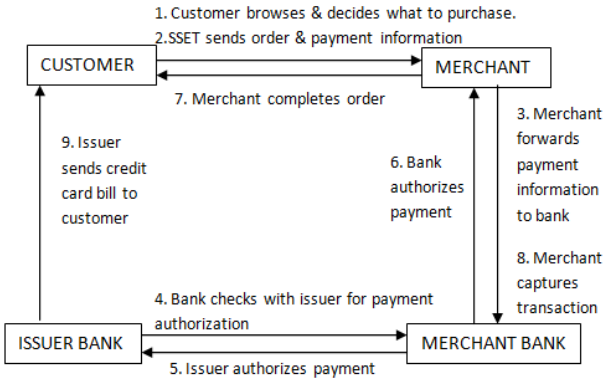

Fig. 1: SET Process.

In SET, five parties are included as cardholder (customer), merchant (web server), acquirer (merchant's bank), payment gateway and issuer (cardholder's bank). The main goal of protocol is to ensure the integrity of payment information, authentication of cardholder as well as merchant and confidentiality of information. For

https://dx.doi.org/10.22161/ijaers/si.16 *ISSN: 2349-6495(P) | 2456-1908(O)*

authentication of cardholders, digital certificates are issued to cardholders, merchants and acquirers by their sponsoring organizations. It also use *dual signature*, which does not allow merchant to access the customer's credit card information, and also hides the order information to banks, to protect privacy.

In SET, message confidentiality and security is provided by cryptography and digital certificate authentication mechanism. In SET, a 56-bit key is randomly generated and message data is encrypted by that key which is further encrypted using the message recipient's public key (RSA). Hence, ―digital envelope‖ of the message is generated. To derive the digital signature, SET uses a distinct public/private key. Each SET participant possess two asymmetric key pairs: first is ―key exchange‖ pair, used for key encryption and decryption, and second ―signature‖ pair for the creation and verification of digital signatures (160-bit message digests).The algorithm used for digital signature creation and verification very strictly follow the property that no two messages will have same message digest and also ensures that any one bit modification will lead to change in half of message digest bits. Thus, approximately, there is negligible, say one in

1,000,000,000,000,000,000,000,000,000,000,000,000,000

,000,000,000 probability that two messages computes

same message digest.

In SET protocol, total 32 messages are transmitted at different frequencies depending upon their purposes. In these messages, the six important messages are PInitReq, PReq, AuthReq, PInitRes, PRes and AuthRes and these are transmitted at highest frequency. Other messages which are used for administrative purposes such as creating certificates, cancelling messages, registration, error handling are transmitted with significantly a lower frequency. A brief overview of how SET transactions are taking place is shown in figure 1.

TABLE.1: SET MESSAGES [8]

Message	***Message meaning***
PInitReq	*Purchase initialization request*
PInitRes	*Purchase initialization response*
PReq	*Purchase request*
Pres	*Purchase response*
AuthReq	*Authorization request*
AuthRes	*Authorization response*

III. OVERLOADS IN SET

SET uses DES for encryption and decryption and the secret key used for DES is again encrypted by RSA mechanism. DES has huge possibility to be easily cracked with the help of modern hardware. Since DES encrypts majority of a SET transaction, **security of DES is a major issue** because the public key cryptography is only used to encrypt secret key for DES keys and for authentication (digital signature) but not for the main body of the transaction.[2][7]

The length of RSA modulus used in SET requires approximately 100,000,000,000MY of computational effort as it is 1024 bits in length. But at the same time, major issue with RSA is its **high computational cost and large message overhead**. Due to ―square and multiply operation‖ and ―simultaneous multiple exponentiation‖ in RSA, one encryption or digital signature generation requires approximately $(1.5\div4)\times|n|$ modulo multiplications computational cost. For instance, one public-key encryption and one digital signature generation is required for PReq generation whose estimated computational cost is recorded as 768 modulo multiplications. Computational cost needed for message generation and verification is given in Table 2 [9].

Digital signatures and public-key encrypted session keys are communication overheads. Also, the 160-bit hash variables contribute to message overhead. The estimated overhead for one digital signature or public-key encrypted session key is |n|. For instance, PReq generation requires one public-key encryption, one digital signature and three hashed variables which combinely costs 2008 bits [8].

IV. PROPOSED PROTOCOL (SSET)

Our motto in designing SSET is to design a highly secure and efficient protocol which fulfills all the requirements of SET as confidentiality, integrity, authentication and non-repudiation.

In traditional SET, RSA encryption mechanism is used to transfer secret key between sender and receiver. Since, computational cost of RSA is very high and also do not have nonce mechanism to ensure freshness and avoid replay attack; hence we propose to use a simple and secure protocol for exchange of session key between two entities by public key cryptography[5][6].

In this protocol, server generates session key and this session key is transferred between the entities securely by encryption with public keys. Once session key is shared, then entities may perform secure communication through shared session key using AES symmetric key algorithm.

4.1 Notations Used:

X->Y: M => X is sending message M to Y

$\{M\}_K$ =>Message M is encrypted with key K K_{XY} => Secret Key between X and Y KR_{alice}=> Private Key of Alice

KU_{alice}=> Public Key of Alice KR_{bob}=> Private Key of Bob KU_{bob}=> Public Key of Bob KR_{server} => Private

https://dx.doi.org/10.22161/ijaers/si.16 ISSN: 2349-6495(P) | 2456-1908(O)

Key of Server KU_{server} => Public Key of Server N_{alice} => Nonce of Alice
N_{bob}=> Nonce of Bob

4.2 Session Key Exchange [1]:

Each participant of SSET will have only one pair of asymmetric key, through which, key exchange as well as digital signature generation and verification will be done.
Protocol is defined as:

1) Alice -> Server: {{A, B, N_{alice}} KR_{alice},}KU_{server}
2) Server -> Alice: {N_{alice}, A, B, $K_{alicebob}$, {{$K_{alicebob}$, A, N_{alice} }KR_{server}}KU_{bob}}KU_{alice}
3) Alice -> Bob: {{$K_{alicebob}$, A, N_{alice}} KR_{server}} KU_{bob}
4) Bob -> Alice: {N_{alice}-1, N_{bob}} $K_{alicebob}$
5) Alice -> Bob: {N_{bob}-1} $K_{alicebob}$

In this protocol, A and B are the identities of Alice and Bob, the communicating parties between whom the secret key needs to be exchanged. S is the server and Alice is the initiator. Alice will generate an initial message consisting id of Alice, id of Bob, and nonce (N_{alice}) of Alice, all encrypted with private key of Alice (KR_{alice}) and again encrypted with public key of Server (KU_{server}) and sends to server S. Server S, on receiving request from Alice, will decrypt this message by its private key (KR_{server}) and Alice public key (KU_{alice}). This encryption mechanism ensures that no one except Server can decrypt Alice message as it is encrypted by public key of Server and hence can only be decrypted by private key of Server.

Now, by decrypting the message, Server will get to know that Alice wants to communicates with Bob, and will give a response which consist Alice current nonce value (N_{alice}), id of Alice, id of Bob, secure session key ($K_{alicebob}$) and a component, whole encrypted with public key of Alice (KU_{alice}). The component consist secret key ($K_{alicebob}$), id of Alice, and its nonce value N_{alice}, encrypted with private key of server and public key of Bob.

Alice will decrypt the message of Server with its private key (KR_{alice}) and extract secret session key $K_{alicebob}$ and sends the component to Bob. Bob will decrypt the received component by its private key (KR_{bob}) and public key of Server (KU_{server}), and extract session key $K_{alicebob}$. Bob will send updated nonce value (N_{alice}-1) of Alice and its nonce N_{bob}, encrypted with session key $K_{alicebob}$ to Alice. Alice will also assures Bob about the session key by sending updated nonce value (N_{bob}-1) to Bob encrypted with secure session key $K_{alicebob}$.

In the whole protocol, authentication is ensured as encryption is done by private key of sender and receiver decrypts it by public key of sender, thereby verifying sender's identity. Further each message is encrypted with public key of receiver to ensure secrecy. With some computational and communication overhead, the protocol is secure in session key generation and exchange. Use of asymmetric key encryption leads to some overheads, but it also ensures secrecy of the session key. Once the session key is shared securely, the symmetric key encryption is used for the actual message transmission.

Traditional SET uses DES for encryption but can very easily be cracked just by following exact reverse operation of encryption. If we use 3DES in place of DES, it will take more time and thus overall efficiency will be decreased. Hence, we proposed a one more updation to SET i.e., use AES which is highly secure and avoids differential and linear cryptanalysis.

4.3 SSET Protocol and Working

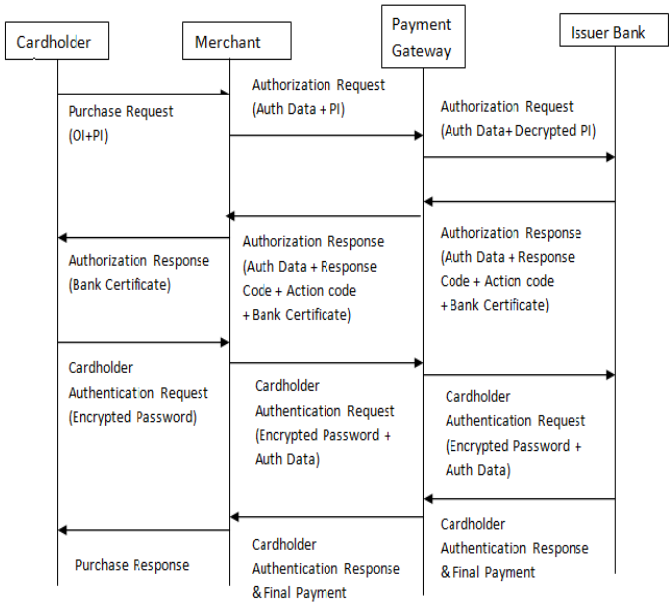

Fig. 2: SSET transaction process.

1) Cardholder browses and selects items to be purchased.
After selecting, they will get a complete order which contains list of items to be purchased. Now cardholder will generate OI (ORDER INFORMATION) and encrypted PI (PAYMENT INSTRUCTION) and will prepare purchase request to be send to merchant.

2) Merchant will now process the OI and send authentication request to payment gateway along with encrypted PI.

3) Payment gateway will decrypt PI and will forward the authorization request to bank.

4) Bank will verify the PI, verify authorization request and will run some issuer control to check if the cardholder is allowed to make this transaction.

5) Then issuer bank will send an authorization

https://dx.doi.org/10.22161/ijaers/si.16 *ISSN: 2349-6495(P) | 2456-1908(O)*

response which contains authorization data, response code (indicates whether authorization request is approved or not), action code(indicates if cardholder is asked to be authenticate by his password and its certificate) and its certificate.

6) This request is further forwarded to merchant. Merchant will check the action code, if its value is –Y‖, means cardholder is authenticated, and then it will send authorization response to cardholder.

7) Now, cardholder will encrypt its password and sends as an authentication request to merchant, which will be forwarded along with authorization data to payment gateway and then to bank. Finally, bank will decrypt and verify password and sends payment response to payment gateway, then to merchant and finally cardholder will get purchase response.

V. THEORETICAL EVALUATION

We have evaluated the proposed protocol on various parameters like time, storage overhead, computational cost, vulnerability etc. theoretically. Following snapshots shows the practical time taken by processor to perform encryption mechanism.

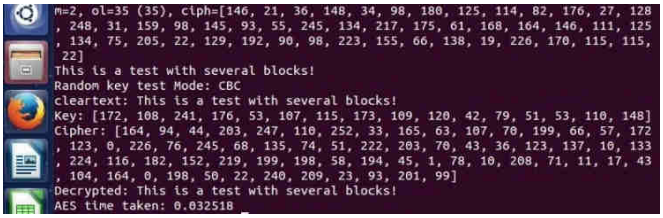

Fig. 3 (a): Performance of AES

Fig. 3 (b): Performance of DES

5.1 Comparison based on various parameters:

SSET and SET are compared on basis of different factors and a comparative study is recorded. Time taken for message encryption in SET (using DES) requires 0.005736 ms whereas SSET messages are encrypted in 0.032518 ms [3][7][8]. Power of any algorithm exist in securing session key and on the basis of our study, session key used in SET can be cracked in 400 days[7] and in SSET, session key will require $5*10^{21}$ years to be cracked. Storage overheads are very high in SET because each participant needs to store two pairs of asymmetric key [4] – one for –key exchange‖ and other for –digital signature generation‖. Also, for authentication purpose, two digital certificates are stored. In SSET, only one pair of asymmetric key is used for key exchange and security is not compromised. Authentication mechanism of SSET also requires only one digital certificate. SET is vulnerable to various attacks such as differential and linear cryptanalysis [8] but SSET is secure against such attacks. Also, session key generation and exchange mechanism is complex in SET but very simple and secure in SSET due to session key exchange protocol.

Hence, it is found that the proposed protocol consumes lesser time than SET for encryption, storage overhead is very less, its computational cost is lower than SET and it is highly secure and simple.

5.2 Computational Cost:

Again, computational cost for message generation and verification is high in SET and less in SSET. Formula for computational cost is= (1.5/4)*|n| mod multiplication, where, |n| is the key size **. [9]**

Let us take an example of **PReq.** Its generation requires one public key encryption and one digital signature. Hence, computational cost (SET) = ((1.5/4)*1024)* 2 = 768 modulo multiplications. Computational cost (SSET) = ((1.5/4)*160)* 2 = 120 modulo multiplications. In the same way, we have calculated computational cost for all six messages generation and verification.

Purchase initialization request (PInitReq) message requires no cost in both SET and SSET as it does not contain any encryption or any digital signature generation and verification. Purchase initialization response (PInitRes) message requires 384 modulo multiplication for message generation as well as verification in SET but requires only 60 modulo multiplication in SSET due to reduced number of encryptions. Purchase response message (Pres) requires 768 modulo multiplication for message generation and 384 modulo multiplication for message verification in SET whereas 120 modulo multiplication for message generation and 60 modulo multiplication for message verification is needed in SSET. Authorization request message (AuthReq) requires 768 modulo multiplication for message generation and 1536 modulo multiplication for message verification in SET whereas only 240 modulo multiplication for message generation and 120 modulo multiplication is required for message verification in SSET. Authorization Response message (Authres) requires 1536 modulo multiplication for message generation and 768 modulo multiplication for message verification in SET whereas SSET needs 60 modulo

https://dx.doi.org/10.22161/ijaers/si.16 *ISSN: 2349-6495(P) | 2456-1908(O)*

multiplication for message generation and 60 modulo multiplication for message verification. Thus computational cost of all six messages is compared and we can conclude that costs have been reduced to a great extent.

VI. CONCLUSIONS

SET has computational and storage overheads due to RSA and DES. Moreover, it is prone to non-repudiation attack. The power of computers is increasing and strong algorithms are required to secure the systems from potential attackers and hackers. Hence, we proposed a upgraded version of SET protocol that uses a simple and secure session key sharing mechanism and also it uses AES for encryption of messages. The proposed protocol is simple, has very low computational and storage overheads and at the same time, it ensures atmost confidentiality, authenticity, integrity and avoids non-repudiation attacks.

REFERENCES

[1] Gupta Anjali and Muzzammil Hussain, "Secure session key sharing using public key cryptography," *Proceedings of the Third International Symposium on Women in Computing and Informatics*.ACM (2015).

[2] Aleisa, Noura,"*A comparison of the 3DES and AES encryption standards,"*International Journal of Security and Its Applications Vol. 9, No. 7, pp: 241-246, 2015.

[3] Sanchez-Avila, C., and R.Sanchez-Reillo,"*The Rijndael block cipher (AES proposal): a comparison with DES*," Security Technology, 2001 IEEE 35th International Carnahan Conference on IEEE, 2001.

[4] Macgegor, Rob,‖Secure electronic transactions: Credit card payment on the web in theory and practice,‖ International Technical Support Organization, 1997.

[5] LLC, SET Secure Electronic Transaction,"SET secure electronic transaction specification," *Book 1: Business Description. Version* 1 (2002).

[6] Li, Yang, and Yun Wang, "Secure electronic transaction (SET protocol)," (2014).

[7] Hamdan.O.Alanazi, B.B.Zaidan, A.A.Zaidan, Hamid A.Jalab, M.Shabbir and Y. Al-Nabhani,"New comparative study between DES, 3DES and AES within nine factors," arXiv preprint arXiv:1003.4085 (2010).

[8] Hanaoka, Goichiro, and Yuliang Zheng, "Improving the secure electronic transaction protocol by using signcryption," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 84(8): pp: 2042-2051, (2001)

[9] Blumenthal, Matt,"Encryption:Strengths and Weaknesses of Public-key Cryptography," CSRS 2007 (2007): 1.

[10] Koponen, A, "*E-commerce, Electronic Payments*," Innovation in Telecommunications 26 (2006).