Journal of Advanced Computer Science & Technology, 6 (1) (2017) 13-19

Journal of Advanced Computer Science & Technology

Website: www.sciencepubco.com/index.php/JACST
doi: 10.14419/jacst.v6i1.6713
Research paper

Enhancing formal specification and verification of e-commerce protocol

Hasan Al-Refai *, Khaldoun Batiha

Philadelphia university
**Corresponding author E-mail: halrefai@philadelphia.edu.jo*

Abstract

Lots of work have been attempted to enhance the SET protocol performance special attention is on E-payment phase. This paper thoroughly analyzes recent works on payment phase; it has been found that this subject requires considerable enhancements, since there are areas, which require further study such as: E-payment phase in SET protocol.
E-payment phase is vast and complex phase it has long series of steps. The behavior of environment is assumed by the phase and is restricted to the rules built by their proposed protocol. This paper will follow Ph-Spi calculus for formalizing and analyzing enhanced payment phase of SET protocol by reducing the number of transactions with many additional operators.
A new agent controller will be formally modeled, which we can rely upon to make automated decisions during interaction with a dynamic protocol environment. So, this agent controller is used to terminate the transaction process in any case of fraud or attack. This paper is conjunction between our previous works of E-payment phase in SET protocol and other works in Ph-Spi calculus in purpose of analyzing and proving the main security properties: authentication and privacy to evaluate the efficiency of the enhanced security of electronic payment phase for SET protocol (E-SET) using Ph-Spi calculus.

Keywords: *SET Protocol; Cryptographic Protocol; SPI and PH-SPI Calculus; Authentication and Privacy.*

1. Introduction

In the past, many significant researchers in e-commerce field have been under gone researching in wide spread of algorithms and techniques for electronic trading on the Internet. The main target for e-commerce users is how to make trade transactions in safely manner and to fulfill individual needs of the participating parties. E-payment becomes the core part of e-commerce when the rapid development of card payment for online purchase of goods kept a side and to prevent wider acceptance safely of electronic transaction becomes a key issue [1], [2], [3], [22], [23], [24].
Merchant and disclosure of credit card number concern users about illegal users of information of third party and the theft of information. The main purpose of merchant and service providers is the authenticity of card user to prevent bad users to use the stolen credit card number to carry out purchases online [1], [2], [3], [4], [27], [28].
After studying different works on various methods applied on SET protocol, it has been found that works are primarily focused on SET protocol for the registration phase and the purchase phase. There have been relatively lesser works in area of payment phase using formal method [1], [2], [3], [25], [26], [27], [28].
Therefore, this paper aims to bridge gap on research by studying the payment phase. Payment phase is very critical to e-commerce; which provides support to maintain confidential data and bank accounts information. Any misuse of such information like hacking, the data can be easily used against the customer and will have deep rooted affect on the performance of system and have a permanent damage to bank reputation. It is important that efficient steps are taken to secure and verify data entry, enabling to achieve trust of clients in the business.
Most of the e-commerce protocols are developed in different methods to be verified. Unfortunately, still suffering for ensuring that as another authentication cryptographic protocols, this is because of e-commerce protocols deals with unknown and dynamic environment that have rapid changes and development of disclosed knowledge to environment [6], [8], [10], [11], [12].
The objective of the study is to use formal methods for payment phase. Current literature lacks research on this area.
First formal method was used to analysis of security protocols applied rarely to key distribution protocols between two parties by Dolev and Yao [9] that able to execute multi-processes of protocol concurrently. Following to their work, researchers develop different tools and methods for analyzing and verifying security protocols; like BAN logic. The development of new languages followed as CAPSL, Pi calculus and Spi-calculus [6], [13], [14], [15], [16], [17], [18], [19].
Pi calculus is considered as the most-recent work in formal method for analysis of e-commerce protocols. Pi calculus' channels are straight forward and powerful, By using it a tool can be created and passed among private channels. Cryptography is used to implement the private channels to protect the data from intruders, but the Pi-calculus does not have such support. By adopting the Pi-calculus channel names to be communicated along with the channels. In this way, it is able to handle concurrent computations where network configuration may vary during the computation. However, the syntax of Pi-calculus does not have the cryptographic operations which are essential for the implementation of channels in distributed systems. Specifically, it does not have any constructors for encryption and decryption [13], [14], [15], [16].
Abadi and Gordon [16] have made some addition in the Pi calculus and named it as Spi-calculus is supportive to cryptographic protocols, which was missing in the original one. The Spi-calculus

enhance the Pi-calculus with primitives for encryption and decryption. Abadi and Fournet [17] have generalized the handling of cryptographic protocols to produce the applied pi calculus.
Spi-calculus is used in the works of (Oras and Boian [18] to verify the Otway-Rees protocol, which is a server, based on providing authentication key for transaction in e-commerce. Adao [19] have further used Spi-calculus to verify and authenticate electronic money. Aszolos and Huszti [20] have used Spi-calculus to authenticate and verify the micropayment method that proposed process for this model for formal protocol, not SET (Secure Electronic Transaction). Bella et al [21] applied inductive method to verify and analyze the purchase phase of the SET protocol.
Previously, we have developed Ph-Spi calculus [14] as an extension of spi calculus. Ph-Spi calculus used to describe and formally verify the main properties of cryptographic protocols. Ph-Spi calculus can be applied for analyzing and verifying a large set of protocols as e-commerce protocols that have a dynamic environment. The language developed to include needed operators to work with such type of protocols.
Ph-Spi calculus will be used to specify the our previous model [3] which used to model an E-payment phase of SET protocol with minimum number of transactions with assuming the use of agent controller.
Ph-Spi calculus contains syntax and semantics to verify SET security protocol properties for the payment phase. The Ph-Spi calculus used in this paper is as the same in our previous work [14].

2. The PH-SPI calculus

This section will provide brief understanding of the syntax and operational semantics of the Ph-Spi calculus. For full description and details, refer to our previous works in [6], [14].].
In Ph-Spi calculus, message is structured to have a tuple of messages as in real case of e-commerce protocols, also includes operators as hash function, timestamp, digital signature, and asymmetric key cryptosystem.
In this paper, we use the Ph-Spi calculus [14] as it is, which contain syntax and semantic to verify SET security protocol properties for payment phase.
The use of formal method in SET protocol found in the literature was lacked. The following summarized the powers of Ph-Spi calculus:

- Representation of security properties, both integrity and secrecy are equivalence.
- Its formal precision.
- Reliance on the powerful scoping constructors of the Spi-Calculus.
- Protocols prevent attacks, and they are developed without explicit specifications for the attacker.

The basic structure of Ph-Spi calculus syntax are built based on names and operators construct as in [14], [6]. Processes, expressions, logical formula and structured messages are modeled as representative structure of Ph-Spi calculus. Based on that all attributes can be expressed to define the driven objects and activities for constructing protocols.
In this paper, we will use the operational semantics presented in [14] where there are two operational semantics the commitment and the reaction relations.
However, the uses of an evaluation function in the operational semantics with its two evaluation modes. One used for Boolean Guards while the other is for expressions.
These two evaluations are expressed as:

- For an evaluation of a Guard

$$\underline{\underline{\cdot}} : \Phi \rightarrow \{tt, ff\}, \text{ is defined by induction on } \Phi.$$

- For an expression

$$\dot{\cdot} : \varepsilon \rightarrow M \cup \{\perp\} \text{ where } \perp \text{ is a distinct symbol } (\perp \notin M)$$

3. E-SET protocols in SPI-calculus

In this section, Ph-Spi-calculus will be used to specify the current model (E-SET) which is placed to describe and formally verify the cryptographic protocols. In the Ph-Spi-calculus, protocols are utilized as processes and properties were proved by using notions of equivalent protocols. Protocol keeps a piece of data X secret by stating that the protocol with X is equivalent to the protocol with X', for every X'.
In this section, we show the standard components Cardholder, Merchant, Issuing bank, Acquiring bank, Payment gateway and ETC (Electronic Transaction Center) which is usually used in payment transactions as modeled in[3]. The model is shown in figure 3.1.
In our previous work [3], we built a model to enhance security of e-payment SET protocol by adding control agent to work as arbitrator.
In this paper, we will show how formally the evaluation function can verify authentication properties through the use of agent controller. However, the two evaluation modes used for expressions and Boolean Guards to validate each of time-stamp, hash function, identification and digital signature.
The purpose of modeling the agent controller as an arbitrator is to increase authentication and decrease non-repudiation by judging the validity of transaction in parallel with the payment flow.
Public, private key and other additional values used in this work to prove the privacy property of protocol. Formally, this can be done by preventing any intruder accesses using the evaluation function in agent controller. It decides to break the content of transaction information operation in case of any attack.

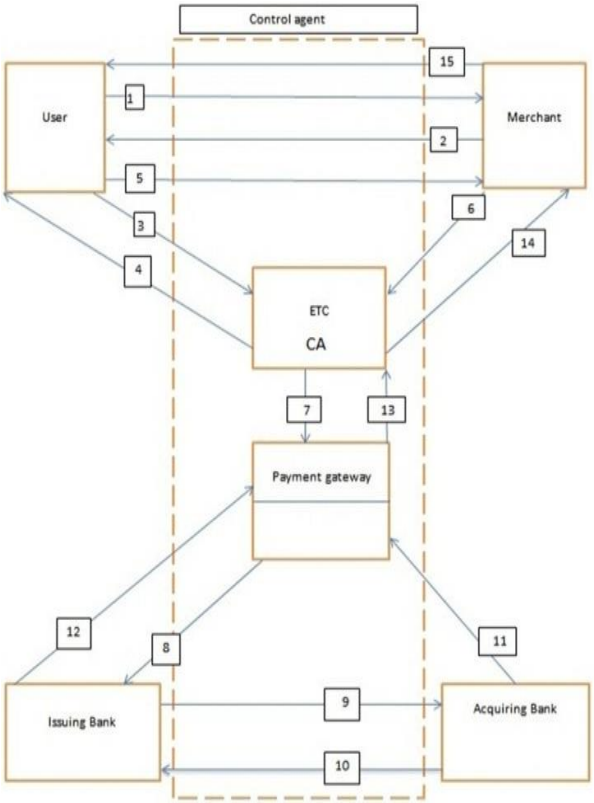

Fig. 3.1: Model Graph.

Now, we will prove how formally the stated transaction between parties in E-SET protocol by Ph-Spi-calculus, firstly start used Ph-Spi-calculus between two parties have transaction between them. It also shown how each party receive or/and send transaction in Ph-Spi-calculus. Table 3.1 give all actions and abbreviations needed in the model with corresponded Ph-Spi calculus symbols.

Table 3.1: Abbreviation

Action	Abbreviation	Symbol
Transaction identity	TID	θ_{id}
Selected item	SI	ξ_i
Certificate of merchant	Mc	o
Certificate of user	Uc	μ
Cost	Ct	δ
Merchant account	Mac	γ
User account	Uac	λ
Time stamp	Ts	ι
Payment information	PI=(Ct, Mac, Uac,TID)	ψ
Order information	OI= (TID, SI, Ct)	χ
User identity	UID	μ_{id}
Merchant identity	MID	o_{id}
Expiration date	E	E
Acquiring bank	A	A
Payment gateway		Θ
Issuing bank		I
ETC		T
Freshly generated variable		η

4. Transaction between cardholder and merchant

Figure 3.2 Ph-Spi calculus, formally stated the transaction between cardholder and merchant as:

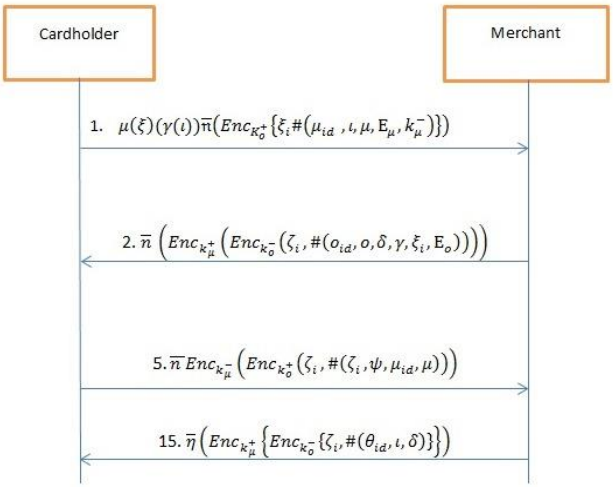

Fig. 3.2: Transactions between Cardholder and Merchant.

Step 1.1: Transaction from cardholder to merchant:

$$|[\nu\iota]|\overline{\eta}\left(\mathrm{Enc}_{K_o^+}\{\zeta_i,\#(\mu_{id},\iota,\mu,E_\mu,k_\mu^-)\}\right)$$

The user sends his/her identity μ_{id} with freshly generated time stamp $|[\nu\iota]|$ and his/her certificate μ, and sending a user secret key k_μ^+ to the merchant and the certificate expiration date E_μ to validate that the amounts will be paid off before defined date E. All of them are hashed by # signed digitally, and encrypted using a secret key of merchant. Then check the message compared with hashed copy of them.
Proof:
The proofs of all transactions will be followed as same as one below. Note that the agent controller will and the evaluation function have same formal function which used to evaluate process to check the validity of the transaction. First the encryption process evaluated as:

$$\overline{\mathrm{Enc}_{K_o^+}\{\#(\mu_{id},\iota,\mu,E_\mu,k_\mu^-)\}}$$
=

$$\begin{cases}\mathrm{Enc}_{K_\mu^+}(\mu_{id},\iota,\mu,E_\mu,k_\mu^-) \text{ if } \pi_i\left(\overline{\overline{\mu_{id}},\iota,\mu,E_\mu}\right)\\ \quad = M\in\mathcal{M} \text{ and } \overline{\overline{K_o^+}} = k\in\mathcal{N}\\ \quad \text{otherwise } \perp\end{cases}$$

Then the evaluation of the hash function is performed as:

$$\overline{\overline{\#(\mu_\iota,\iota,\mu,E_\mu,k_\mu^+)}} = \begin{cases}\text{if } \pi_i\left(\overline{\overline{\mu_\iota,\iota,\mu,E_\mu,k_\mu^+}}\right)\in\mathcal{N} \text{ and } (.)\neq(.)^{-1}\\ \perp, \text{otherwise}\end{cases}$$

That means hashing is perfect and non-invertible.
Then, to evaluate the time stamp the agent controller will ensure that it is a valid and did not change during the running process.

Let $\quad \iota: P\begin{cases}\text{tt if } \iota\in\mathcal{N}\neq\perp\\ \text{ff otherwise}\end{cases}$

Where P is a process in which the transaction running in.
The date expiration for user certification will be checked through the agent controller to validate that the amounts will be paid off before defined date E.
The merchant will receive:

$$\eta(x_1,x_2,x_3,x_4,x_5):$$

Then inputs shows the substitution σ, which is function of a finite partial map to perform substitution of a set of names $(x_1,x_2,x_3,x_4,x_5)\in\mathcal{N}$ to a set of messages $(\xi_1,\xi_2,\xi_3,\xi_4,\xi_5)\in\mathcal{M}$:

$$\sigma\Rightarrow\{{}^{\mu_{id}}/_{x_1},{}^{\iota}/_{x_2},{}^{\mu}/_{x_3},{}^{E_\mu}/_{x_4},{}^{k_\mu^-}/_{x_5}\}$$

Proof
Such set of messages and the fresh set of names are evaluated through the agent controller as:
First, check the structure of the received messages:

$$\zeta_i\langle\mu_{id},\iota,\mu,E_\mu,k_\mu^-\rangle =$$
$$\begin{cases}\langle\mu_{id},\iota,\mu,E_\mu,k_\mu^+\rangle \text{ if } \overline{\overline{\mu_{id}}} = \xi_1 \text{ and } \bar{\iota}=\xi_2 \text{ and } \bar{\mu}=\xi_3\\ \quad \text{and } \overline{\overline{E_\mu}}=\xi_4 \text{and } \overline{\overline{k_\mu^-}}=\xi_5 \text{ for } \xi_1,\xi_2,\xi_3,\xi_4,\xi_5\\ \quad \perp, \text{otherwise}\end{cases}$$

Then the received encrypted messages and the substitution function will be evaluated by agent controller as:

$$\text{let } \langle z_1,z_2,z_3,z_4,z_5\rangle = \langle\xi_1,\xi_2,\xi_3,\xi_4,\xi_5\rangle \text{ in } \phi$$

$$= \begin{cases}\text{tt } \phi\left\{{}^{\xi_1}/_{x_1},{}^{\xi_2}/_{x_2},{}^{\xi_3}/_{x_3},{}^{\xi_4}/_{x_4},{}^{\xi_5}/_{x_5}\right\} \text{ if } (\xi_1,\xi_2,\xi_3,\xi_4,\xi_5)\\ \quad \text{ff}, \text{otherwise}\\ \quad = M_i\in\mathcal{M}\end{cases}$$

The agent controller also must check if mapping avoid capture of names by binders. Substitutions are applied to this transaction process expression and guards in straightly as process in

$$\left\{{}^{\xi_1}/_{x_1},{}^{\xi_2}/_{x_2},{}^{\xi_3}/_{x_3},{}^{\xi_4}/_{x_4},{}^{\xi_5}/_{x_5}\right\}$$

Replaces all free occurrence of x_i by ξ_i possibly renaming bound names in this process avoiding name capture.
Step 1.2: Transaction from merchant to cardholder.

$$\overline{\eta}\left(Enc_{k_\mu^+}\left(Enc_{k_o^-}(\zeta_i,\#(o_{id},o,\delta,\gamma,\xi_i,E_o))\right)\right)$$

The merchant responds by sending back to user the merchant identity o_{id} , merchant certificate o , total cost of selected items δ,

merchant account γ, list of items ξ_i and the expiration date at merchant certificate E_o.
All messages ζ_i are singed digitally and hashed with encryption of a merchant secret key then encrypted with a user a public key. The proof almost exactly as in proof of step 1.1.
Where the sending transaction tuple takes the form of

$$\overline{\eta}\left(Enc_{k_\mu^+}\left(Enc_{k_o^-}(\zeta_i, \#(\zeta_1, \zeta_2, \zeta_3, \zeta_4, \zeta_5, \zeta_6))\right)\right).$$

In the response to the transaction from merchant to user then user will receive $\eta(x_i, (x_1, x_2, x_3, x_4, x_5, x_6))$.
Same as in step 1.1, the substitution σ take as

$$\sigma \Rightarrow \left(\zeta_1/_{x_1}, \zeta_2/_{x_2}, \zeta_3/_{x_3}, \zeta_4/_{x_4}, \zeta_5/_{x_5}, \zeta_6/_{x_6}\right).$$

The proof almost exactly as in step 1.1.
Step 1.3: Transaction from cardholder to merchant

$$\overline{n}\left(Enc_{k_\mu^-}\left(Enc_{k_o^+}(\zeta_i, \#(\zeta_i, \psi, \mu_{id}, \mu))\right)\right)$$

The user will respond by sending the list of items, the payment information that contain: cost of selected item, merchant accountγ, user account represents credit card number λ and the transaction identity θ_{id}.
Note, we compose $\delta, \gamma, \lambda \; and \; \theta_{id}$ in one digest message ψ for more securing the message which containing the account numbers and the transaction identity.
The merchant will respond by receiving:

$$n(x_i, (x_1, x_2, x_3, x_4, x_5, x_6))$$

Same as in step 1.1, then substitution σ take action as:

$$\sigma \Rightarrow \{\left(\zeta_1/_{x_1}, \zeta_2/_{x_2}, \zeta_3/_{x_3}, \zeta_4/_{x_4}\right)\}$$

The proof almost exactly as in step 1.1.
Step 4: Transaction from merchant to cardholder

$$\overline{\eta}\left(Enc_{k_\mu^+}\left\{Enc_{k_o^-}\{\zeta_i, \#(\theta_{id}, \iota, \delta)\}\right\}\right)$$

This merchant send the identity θ_{id} and the time stamp after merchant checked it with the previously received from the user to insure authentication of the transaction and privacy of the whole sent items by checking the equivalency of ι in every transaction. The same process of evaluation will be performance for cost of selected items to ensure there is no change or modifications done, all sent by all sent by merchant singed digitally and encrypted within a merchant public key k_o^- then re-encrypted with the user secret key k_μ^+.
Proof:
To evaluate the time-stamp ι and for cost of selected items δ, the equality defined in the Boolean Guard evaluation function [14] as $\overline{[\iota_\mu : \iota_o]}$, in this case the agent controller (evaluation function) will validate the equality for both transactions of freshly generated time stamp ι_μ and the earlier time stamp sent by merchant to the user ι_o. Otherwise the transaction will be terminated.
For the cost of selected items sent by user δ_μ and the one returned by merchant δo evaluated as equal in the same way as above. Hash function applied for all of the sent elements $\#\zeta_i$ for I = {1, 2… n} we have $(\zeta_1, \zeta_2, \zeta_3, \dots\dots\zeta_n)$ where n is the total number of the sent elements $(\theta_{id}, \iota, \delta)$ the proof of hash function it exactly same as in step 1.
To prove the privacy property for a sent messages, the process of encryption applied for these messages without disclosing any items to environment. Therefore, the use of Boolean guard in evaluation function as in Step1 can be followed to evaluate the process of encryption/ decryption. This to ensure that there is no chance for any intruder to intercept the transaction. This can lead to explicitly guarantee the authentication property since that only authorized parties can be involved in each transaction.

5. Transaction between cardholder and ETC

Figure 3.3 stated transaction between cardholder ETC by the following:

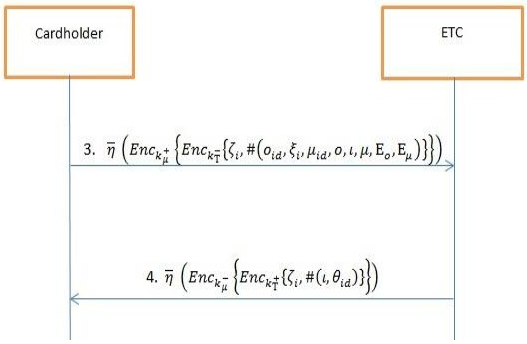

Fig. 3.3: Transactions between Cardholder ETC.

Step 2.1: Transaction from cardholder to ETC

$$\overline{\eta}\left(Enc_{k_\mu^+}\left\{Enc_{k_T^-}\{\zeta_i, \#(o_{id}, \xi_i, \mu_{id}, o, \iota, \mu, E_o, E_\mu)\}\right\}\right)$$

In this transaction, that user send the merchant identity o_{id} , list of selected items ξ_i,where i={1,2,……,n} to denote what number of selected items is to purchase, these are sent with the time stamp and the user identity μ_{id}, merchant certificate o and it's expiration date E_o and user certificate μ with it's expiration date E_μ .
Proof:
By the induction hypothesis on applying a Boolean Guard and expression evaluation functions we can explicitly guarantee the authentication and privacy properties
Step 2.2: Transaction from ETC to cardholder

$$\overline{\eta}\left(Enc_{k_\mu^-}\left\{Enc_{k_T^+}\{\zeta_i, \#(\iota, \theta_{id})\}\right\}\right)$$

The ETC will response by sending the time stamp and the transaction identity.
Proof:
Proof explicitly can be done as in step 1. 4, using the equality Boolean Guard in evaluation function for each element sent by transaction can be evaluated to have a same as the received element. Through that the time stamp and transaction identity sent by user is evaluated to have a same as the received by the ETC. otherwise the transaction will be terminated. This can lead us to guarantee the authentication and privacy properties.

6. Transaction between ETC-merchant

Figure 3.4 stated transaction between merchant and ETC by the following:

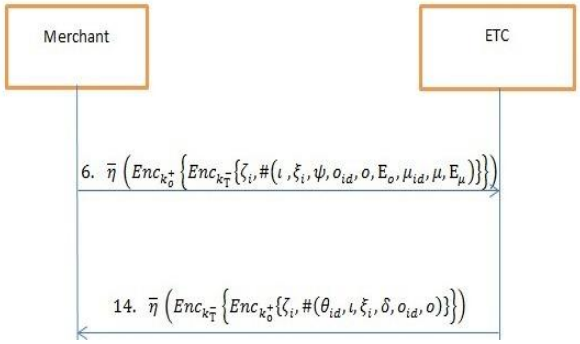

Fig. 3.4: Transaction between Merchant and ETC.

Step 3.1: Merchant-ETC transaction

$$\bar{\eta}\left(Enc_{k_o^+}\left\{Enc_{k_T^-}\{\zeta_i,\#(\iota,\xi_i,\psi,o_{id},o,E_o,\mu_{id},\mu,E_\mu)\}\right\}\right)$$

In this transaction the merchant send the time-stampι, and the list of selected items ξ_i , a digest message ψ for payment information which compose all of $\delta,\gamma,\lambda,\theta_{id}$ and merchant identity o_{id} , merchant certificate o with its expiration date E_o , user identity μ_{id} , user certificate μ with its expiration date E_μ .
Proof:
Explicitly same process of evaluation for proving authenticity and privacy properties can be followed as in earlier proofs.
Step 3.2: ETC- Merchant transaction

$$\bar{\eta}\left(Enc_{k_T^-}\left\{Enc_{k_o^+}\{\zeta_i,\#(\theta_{id},\iota,\xi_i,\delta,o_{id},o)\}\right\}\right)$$

The ETC responses for received message from both the user in (step 1.3) and from the payment gateway.

7. Transaction between ETC and payment gateway

Figure 3.5 stated transaction between ETC and payment gateway by the following:

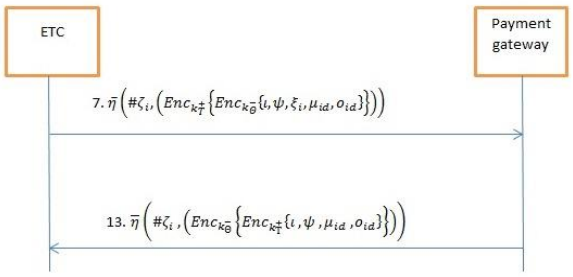

Fig. 3.5: Transaction between ETC and Payment Gateway.

Step 4.1: ETC-payment gateway transaction

$$\bar{\eta}\left(\#\zeta_i,\left(Enc_{k_T^+}\left\{Enc_{k_\Theta^-}\{\iota,\psi,\xi_i,\mu_{id},o_{id}\}\right\}\right)\right)$$

The ETC after receiving the three transactions from the user and the merchant in (3, 4, 6 as in figure 3.1) will be ready to send a payment request to the payment gateway which contains: the list of selected items ξ_i , payment information ψ, time stamp ι , user identity μ_{id} , merchant identity o_{id} . All are hashed and digitally signed with the ETC public key.
Proof:
Explicitly can be followed as in previous proofs by applying the evaluation function for validating each message carried in the transaction.

At the other side, the payment gateway will evaluate the received transaction by evaluating each message and check it is valid or not to decide whether to proceed to next transaction or terminate the processes.
Step 4.2: Payment gateway- ETC
As shown in figure 3.1, the payment gateway will respond to ETC after sending the received transaction to the issuing bank (step 8) and waiting reply from both issuing bank (step 12) and acquiring bank (step 11) by sending:

$$\overline{\eta}\left(\#\zeta_i,\left(Enc_{k_\Theta^-}\left\{Enc_{k_T^+}\{\iota,\psi,\mu_{id},o_{id}\}\right\}\right)\right)$$

The payment gateway will send back what had been received as response to ETC without selected items ξ_i all are encrypted and signed digitally by his public key. The evaluation function will verify the equivalency of received tuple of message to the tuple previously sent by Payment gateway. Proof, same as in previous steps.

8. Transaction between payment gateway and issuing bank

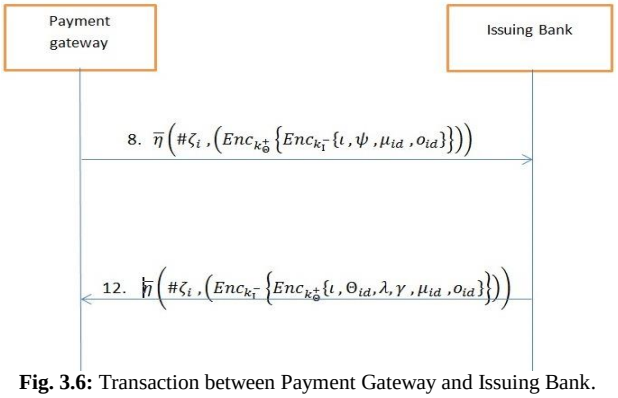

Fig. 3.6: Transaction between Payment Gateway and Issuing Bank.

Step 5.1: Transaction between payment gateway and issuing bank
The payment gateway will verify the received transaction from the ETC then send the issuing bank in step 8 (as in figure 3.1) fund requesting from the user as:

$$\overline{\eta}\left(\#\zeta_i,\left(Enc_{k_\Theta^+}\left\{Enc_{k_I^-}\{\iota,\psi,\mu_{id},o_{id}\}\right\}\right)\right)$$

The sent messages from payment gateway are: time stamp, payment information ψ , user identity μ_{id} , and merchant identity o_{id} .
The issuing bank will receive this transaction, verify the validity at it, and send a request of withdrawing money for the merchant to the acquiring bank step 9 (as in figure 3.1).
Proof:
The proof explicitly can be easily followed as in previous steps.
Step 5.2: Issuing bank- payment gateway transaction
The issuing bank will send what received from acquiring bank for deposit of money to the payment gateway.

$$\overline{\eta}\left(\#\zeta_i,\left(Enc_{k_I^-}\left\{Enc_{k_\Theta^+}\{\iota,\Theta_{id},\lambda,\gamma,\mu_{id},o_{id}\}\right\}\right)\right)$$

The issuing bank will digitally sign the encrypted message with a hashed one by its secret key, the message contain the time stamp ι, transaction identity Θ_{id} , user account λ, merchant account γ, user identity μ_{id} , and merchant identity o_{id} .
To validate the message elements, the evaluation function as an agent controller will be used.
Proof:
As in step 1.1, explicitly the evaluation function will ensure the correctness and completeness for the transaction depending on the collected knowledge gained by agent controller from each element.

Already the agent controller has enough knowledge of each element of the transaction that makes the evaluation easier to apply. The evaluate function to validate and decide the correctness and completeness of the transaction. The evaluation can be explicitly followed as same as in step 1.1.

9. Transaction between issuing bank and acquiring bank

Figure 3.7 stated transaction between issuing bank and acquiring bank by the following:

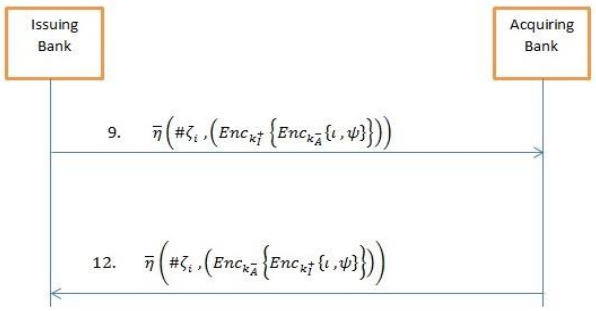

Fig. 3.7: Transactions between Issuing and Acquiring.

Step 6.1: Acquiring bank- issuing bank transaction
The acquiring bank will deposit the amount money specified in step 9 to be deducted from the user account, and send back the time stamp ι and the payment information ψ to the issuing bank. All are hashed and digitally signed using acquiring bank secret key:

$$\overline{\eta}\left(\#\zeta_i, \left(\mathrm{Enc}_{k_{\bar{A}}^-}\left\{\mathrm{Enc}_{k_I^+}\{\iota, \psi\}\right\}\right)\right)$$

Step 6.2: Issuing bank- acquiring bank transaction
After what issuing bank had received from the payment gateway in step 8. It will be ready to send the time stamp with the payment information which contain of the cost of selected items ξ and merchant account γ, user account λ and transaction identity Θ_{id}. All digitally singed and hashing using issuing bank public key:

$$\overline{\eta}\left(\#\zeta_i, \left(\mathrm{Enc}_{k_I^+}\left\{\mathrm{Enc}_{k_{\bar{A}}^-}\{\iota, \psi\}\right\}\right)\right)$$

Proof:
The agent controller will use the evaluation function to verify and validate the elements of transaction. The proof explicitly can be smoothly followed as in step 1.1.

10. Transaction between acquiring and payment gateway

Figure 3.8 stated transaction between acquiring bank and payment gateway by the following:

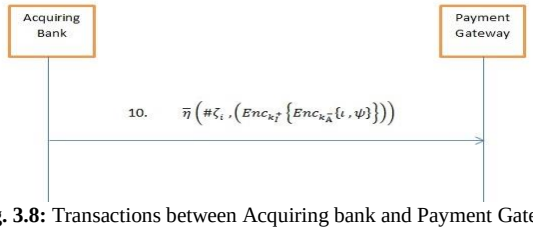

Fig. 3.8: Transactions between Acquiring bank and Payment Gateway.

Step 7.1: Acquiring bank- payment gateway transaction
As in step 8 (figure 3.1), where the payment gateway already sent the user and merchant identity with time stamp, the same transaction will be send by the acquiring bank to the payment gateway. At that point the payment gateway can evaluate the both transactions to validate that no changes or any type of interference from any outsider party (intruder). The transactions send hashed and digitally signed by the acquiring bank public key as:

$$\overline{\eta}\left(\#\zeta_i, \left(\mathrm{Enc}_{k_{\bar{A}}^+}\left\{\mathrm{Enc}_{k_{\bar{\Theta}}^-}\{\iota, \psi\}\right\}\right)\right)$$

11. Conclusion

In this paper, we have proved E-SET in Spi-calculus, it also apply for each transaction between parties. Controller agent evaluates each transaction, check validity and evaluation function to validate and decide the correctness and completeness of transaction .consequently, increase authentication and privacy properties and decrease non-repudiation.

References

[1] Singh, S, (2009). "Emergence of payment systems in the age of electronic commerce: The state of art," Internet, 2009. AH-ICI 2009. First Asian Himalayas International Conference on, vol., no., pp.1-18, 3-5 Nov. 2009.
[2] Zhang Xuan, Huang Qinlong and Peng Peng, (2010). "Implementation of a suggested E-commerce model based on SET protocol". Eighth ACIS International Conference on Software Engineering Research, Management and Applications. IEEE, 2010.
[3] Hasan Al-Refai, Ali Alwaneh, Khaldoun Batiha, Ahmad Jarah. (2014). Enhanced model of Payment Phase for SET Protocol. International Journal of Video&Image Processing and Network Security IJVIPNS-IJENS Vol: 14 No: 0 1.
[4] A.A.Koponen. (2006). E-COMMERCE, ELECTRONIC PAYMENTS. http://www.cse.tkk.fi/f i/opinnot/T-109.7510/2006/E-commerce.pdf.
[5] Anssi Mattila. SET & SSL: IS THERE A COMPARISON FOR A GOOD NIGHT SLEEP. International Journal of Enterprise Computing and Business Systems. Volume 2 Issue 2 July 2012.
[6] Al-Refai (2009) Evaluation Technique in the Spi-Calculus for Cryptographic Protocols. Third International Symposium on Innovation in Information Communication Technology- ISIICT.
[7] Laudon Kenneth C. and Traver Carol Guercio, (2005). "E-Commerce, Business. Technology". Society. Second edition.
[8] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The Spi calculus. Information and Computation, 148(1):1–70, 1999. https://doi.org/10.1006/inco.1998.2740.
[9] D. Dolev and A. C. Yao. On the security of public key protocols. IEEE Transactions on Information Theory, 29(2):198–208, 1983. https://doi.org/10.1109/TIT.1983.1056650.
[10] G. Bella, F. Massacci, L. C. Paulson, and P. Tramontano. Formal verification of cardholder registration in SET. In Proc. 6th European Symposium on Research in Computer Security (ESORICS), volume 1895 of LNCS, pages 159–174. Springer, 2000. https://doi.org/10.1007/10722599_10.
[11] C. Meadows and P. Syverson. A formal specification of requirements for payment transactions in the SET protocol. In Proc. 2nd Financial Cryptography Conference (FC), volume 1465 of LNCS, pages 122–140. Springer, 1998. https://doi.org/10.1007/BFb0055477.
[12] M. Abadi and A. Gordon. A bisimulation method for cryptographic protocols. Nordic Journal of Computing, 5(4):267–303, 1998. https://doi.org/10.1007/bfb0053560.
[13] R. Milner Communicating and Mobile System: the pi-Calculus. Cambridge University Press, 1999.
[14] Hasan Al-Refai, Khaldoun Batiha, Ali Alwaneh, Saleh Bani Hani. (2014). Improved SPI Calculus for Reasoning on Cryptographic Protocols. International Journal of Video&Image Processing and Network Security IJVIPNS-IJENS Vol:14 No:01
[15] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The Spi Calculus. Information and Computation, 148(1):1–70, January 1999. Full version available as SRC Research Report 149, January 1998.

[16] M. Abadi and A. Gordon. Reasoning about cryptographic protocols in the Spi Calculus. In A. Mazurkiewicz and J. Winkowski, editors, Proceedings of the 8th International Conference on Concurrency Theory (CONCUR), volume 1243 of Lecture Notes in Computer Science, pages 59–73, Warsaw, Poland, July 1997. Springer. https://doi.org/10.1007/3-540-63141-0_5.
[17] Martin Abadi and Cedric Fournet. “Mobile values, new names, and secure communication”. POPL 2001.
[18] Horea and Florian Boian. OrosSpi calculus analysis of Otway-Rees protocol. International Journal of Computers, Communications & Control (IJCCC) 3:427-432 • January 2008.
[19] Pedro Adao. "Electronic Money within My-Calculus". Applied Mathematics and Computation Diploma Thesis. July 2002.
[20] Laszlo Aszalos and Andrea Huszti. "Applying Spi-Calculus for PayWord". Eighth International Conference on Applied Informatics. 2010.
[21] Giampaolo Bella, Fabio Massacci and Lawrence C Paulson. Verifying the SET Purchase Protocols. Kluwer Academic Publishers. Printed in the Netherlands. 2005.s
[22] Zihao Shen; Hui Wang, (2010). "An improved SET protocol payment system," Computer and Communication Technologies in Agriculture Engineering (CCTAE), 2010 International Conference On , vol.1, no., pp.400-403, 12-13 June 2010.
[23] Li Yabo, (2008). "The Design of the Secure Electronic Payment System Based on the SET Protocol," ICCSIT, pp.29-33, 2008 International Conference on Computer Science and Information Technology. https://doi.org/10.1109/ICCSIT.2008.175.
[24] Fei Xu, Zhang Ai Ming and Liang Wan, (2010). "Formalizing and Checking SET Protocol Based on TLA." IEEE, 2010.
[25] Hassan M. Elkamchouchi 1, Eman F. Abu Elkhair2 and Yasmine Abouelseoud3. AN IMPROVEMENT TO THE SET PROTOCOL BASED ON SIGNCRYPTION.. International Journal on Cryptography and Information Security (IJCIS), Vol.3, No. 2, June 2013.
[26] Y. LI AND Y. WANG, "SECURE ELECTRONIC TRANSACTION (SET PROTOCOL)", [ONLINE] AVAILABLE AT HTTP://WWW .PEOPLE.DSV. SU.SE/~MATEI/COURSES/IK2001_SJE/LI-WANG_SET.PDF 2012.
[27] Boping Zhang, and Shiyu Shang, (2009). An Improved SET Protocol. Proceedings of the 2009 International Symposium on Information Processing (ISIP’09) Huangshan, P. R. China, August 21-23, 2009, pp. 267-272.
[28] Xu Yong and Liu Jindi, (2010). "Electronic Payment System Design Based on SET and TTP," ICEE, pp.275-278, 2010 International Conference on E-Business and E-Government, 2010. https://doi.org/10.1109/ICEE.2010.77.