The Formal Verification and Improvement of Simplified SET Protocol

Meihua Xiao[1, 2]
1 School of Software, East China Jiaotong University, Nanchang 330013, China
2 State Key Laboratory for Novel Software Technology, Nanjing University, 210093, China
Email: xiaomh@ecjtu.edu.cn

Zilong Wan[3] and Hongling Liu[3]
3 Computer Center, Nanchang University, Nanchang 330031, China
Email: {ncuwzl, nculhl}@126.com

***Abstract*—Model checking has been successfully applied to verify the security properties of network protocols. In this paper, we propose a formal modeling method using the PROMELA language based on simplified model of SET payment protocol proposed by Lu & Smolka, and use LTL to describe authentication property. Under the hypothesis of the network environment being controlled by the intruder, we use SPIN to find the attacks and improve the verification efficiency by using the optimization strategies of atomic steps and Bit-state hashing technology. Finally, we improve the existing vulnerabilities of the SET protocol.**

***Index Terms*—Model checking, SET protocol, Payment process, Vulnerabilities.**

I. Introduction

With the development of e-commerce, people have paid more attention to its safety. The safety of e-commerce transaction mainly depends on password techniques, security mechanism and security protocols. The emergence of e-commerce protocols makes e-commerce transaction having higher security. The SET protocol is one of main security protocols used in e-commerce. It is used in an open network environment, which is based on the process of credit card transaction. It provides the authentication among consumers, merchants and banks to ensure that the confidentiality and reliability and non-repudiation of transaction.

Because the security problem of protocol is very subtle, the vulnerabilities of some uncomplicated protocols need a long time to be found. Therefore, people pay more attention to the security of SET protocol, and analyze the security of SET protocol to find the potential safety hazard or prove its safety. It is of great significance for the further promotion application and development of SET protocol, Therefore, many experts and scholars try to analyze and study on SET protocol from different points of view using different methods.

The SET protocol consists of five phases. The first two phases are used by the agents participating in the protocol to register their keys and get the appropriate certificates. The remaining phases constitute the electronic transaction itself. They are also the focus of our research.

Lu& Smolka protocol is a famous simplified version of the SET payment protocol. In Ref.[1], they used the model checker FDR to validate the protocol, and found that Lu& Smolka protocol is not safe in an open network environment. In Ref.[2], they used the symbolic model checker SMV to validate the SET payment protocol, and gave out its existing attacks. In this paper, we use the tool of SPIN to analyze the simplified SET protocol, and improve the protocol. They are significantly different in the aspect of modeling with the two references above.

II. Model Checking

Model checking [3] is an automated analysis and verification techniques for verifying the finite state system, which is an important method in formal verification. A protocol model of finite state system is constructed based on the algebraic method, and the security of the protocol is analyzed using tools of states monitoring. Its basic idea is to use state transition system (S) and modal/temporal logic formula (F) to express the system behavior and describe property of the system respectively [4]. So, whether the system has the desired properties or not is converted to a mathematical problem that whether state transition system(S) is a model of formula (F) or not. Namely, we need judge that S|=F is true or not. If not, the counter example is given.

A. Model Checking Tools

At present, model checking tools are mature. They can be mainly divided into the FDR based on the CSP and automaton, the SPIN based automatic machine, symbolic model checking tool SMV and software model checking tool Java Path Finder, etc. SPIN [5] is a famous tool of analyzing and verifying the logic consistency on

Manuscript received November 23, 2013; revised January 6, 2014; accepted February 18, 2014.
The work is supported by the National Natural Science Foundation of China(No.61163005),Natural Science Foundation of Jiangxi, China (No.2010GZS0150,20132BAB201033),Science and Technology College in Jiangxi Province Ground Project (No. KJLD13038), State Key Laboratory for Novel Software Technology Open Project (No.KFKT2012B18).
Corresponding author: Meihua Xiao(Email: xiaomh@ecjtu.edu.cn)

doi:10.4304/jsw.9.9.2302-2308

concurrent system. It has been successfully used in the security protocol verification, the control system verification, software verification, the optimization planning and other fields. It can also be used to test the security of the network protocol design and report the deadlock, invalid loop and inaccessible state in the system.

B. The Working Mechanism of SPIN

After inputting the protocol system Described using PROMELA language, SPIN can execute random simulation of system, also generate the corresponding C program, and check the state space of system using exhaustivity or part search.

The basic structure of the SPIN model checker is illustrated in Figure 1.

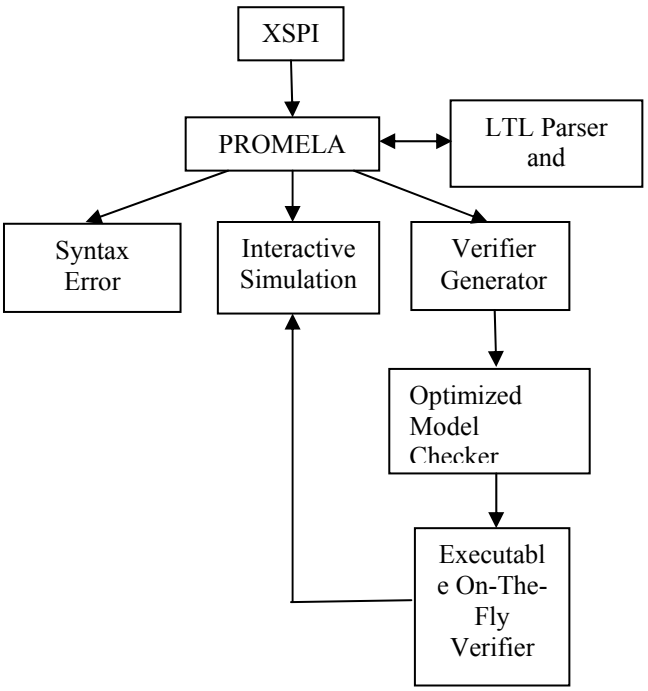

Figure 1. The structure of SPIN simulation and verification

It starts with the specification of a high level model of a concurrent system or distributed algorithm based on graphical front-end XSPIN of the SPIN. After checking the syntax errors, interactive simulation is performed until the design of the system model can behave as we had expected [6] [7]. Then, it will generate an optimized on-the-fly verification program from the high level specification. The program will be executed after being compiled by a verifier. If any counter examples against the correctness are detected, it will go back to the state of interactive simulation and inspect in detail to determine the causes of emerging the counter examples.

C. PROMELA Language

SPIN uses PROMELA (Process Meta Language) as modeling language. PROMELA is a language for building verification models that represent an abstract of a system, which contains only those aspects that are relevant to the properties one wants to verify [8]. A PROMELA program consists of processes, message channels, and variables. Processes are defined globally, while message channels and variables can be declared either globally or locally within a process. Processes are used to specify system behaviors, and channels and global variables are used to define the environment in which the processes run.

III. THE FORMAL MODELING OF SET PROTOCOL

SET protocol is known as security electronic transaction protocol. It is proposed jointly by the companies of Master Card, Visa, Netscape, and Microsoft and so on in June 1, 1997. It is a kind of new electronic payment mode. SET is designed to solve the credit card transaction among the consumers, merchant and banks and ensure the integrity of transaction data, non-repudiation of transaction and legal status of related principals.

A. Formal Analysis and Description of The Protocol

The SET protocol is composed of 17 sub- protocols. It covers all the processes of certificate management and payment system. Taking into account the complexity of the SET protocol itself, we will select the core components of the protocol, such as purchase request, payment certification and get the payment, as the research objects. For the convenience of description, some symbols are defined firstly:

OI: Order information. Omitting some detail data, it mainly includes the transaction number TransId and the purchase amount PA of the customer.

PY: The amount that customers are willing to pay. It may be lower than PA if the customers are trying to cheat.

PI: Payment instruction. Omitting some detail data, it mainly includes TransId, PY and CA.

CA: The account information of customers

MA:The account information of merchants

Sx(Y): The signature for the message Y using private key of the principal X.

Ex(Y): The encryptions for the message Y using public key of the principal X.

The payment process can be described by the following steps:

Step1:Initiate Request: The cardholder C sends message 1 to the merchant M to ask for a transaction number.

Step2: Initiate Response: The merchant M assigns a unique identifier to the transaction, after being signed, it will be returned to the cardholder C as the message 2.

Step3: Purchase Request: The cardholder C checks and accepts the message 2, and produces the message 3 that be sent to the merchant M. Where OI can be interpreted by merchant C and PI can be interpreted by the payment gateway.

Step4: Authorization and Capture Request: The merchant M checks and accepts the message 3, and produces the message 4 that be sent to the payment gateway.

Step5: Authorization and Capture Response: The payment gateway checks and accepts message 4, identifies the payment card of the cardholder C, checks the consistency of between the OI and PI, and produces message 5 that will be sent to the merchant M.

Step6: Purchase Response: The merchant M checks and accepts the message 5, and responses to the cardholder C by sending the message 6 which is getting from the message 5.

After the analysis of the payment process, we can acquire the abstract model of the process. As is shown in figure 2. The formal representation of the payment process is as follows:

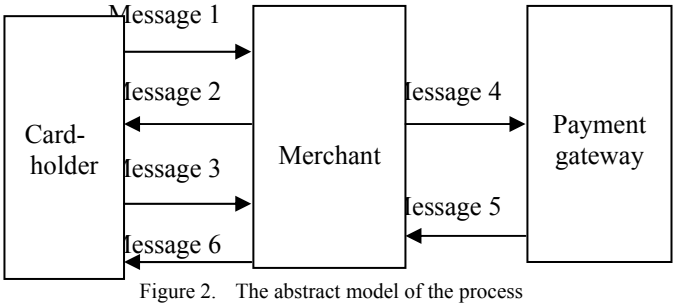

Figure 2. The abstract model of the process

Step 1: C->M: <TransId_Req>
Step 2: M->C: <Sm(TransId)>
Step 3: C->M: <Sc(TransId),Em(OI),Sc(Ep(PI))>
Step4:M->P:
<Sc(Ep(PI),Sm(TransId),Ep(TransId,AA,MA))>
Step 5: P->M: <Sp(TransId,TrRes)>
Step 6: M->C: <Sp(TransId,TrRes)>

B. Protocol Modeling

Through the analysis, we know that the payment process of the SET protocol includes four abstract principals: three legal principals (the cardholder C, the merchant M and the payment gateway P) and the intruder I. There are two kinds of message channels, one is used for transmitting messages between the cardholder and the merchant, the other is used for transmitting messages between the merchant and the payment gateway. The statements of message channels are as follows.

```
chan c_m=[0]of{mtype,mtype,mtype,mtype,mtype,mtype,
                mtype};
chan m_p=[0]of{mtype,mtype,mtype,mtype,mtype,mtype,
                mtype};
```

The resulting set of names we will use is defined as follows:

```
mtype={C,M,P,I,Ckey,Mkey,Ikey,Nm,Ng,TNo,temp};
```

The principals of the protocol include cardholder C, merchant M, payment gateway P, and intruder I. From this, we model based on the three legal principals and the one illegal principal. For the cardholder, it selects a communication object firstly, sends its initiate request secondly, receives the initiate response thirdly, and checks the secret keys finally. If the secret keys are matched with each other, then go on the trade, otherwise exit the trade. After the trade is successful, the value parameter flagC is set as 1. We can acquire the modeling process of the merchant and the payment gateway similarly.

One key principal in the modeling process is the intruder. It has powerful ability of attack, this is also we are interested in. The analyses on the intruder's ability are as follows:

(1) The intruder knows all the principals and their public keys, and it has its own public key and private key.
(2) The intruder can resend and modify the messages that it steals from others.
(3) The intruder can pretend to be legal principal and communicate with other principals.
(4) The intruder can generate new messages based on the known knowledge and send them out.

The modeling process of the four principals can be expressed as the figure 3.

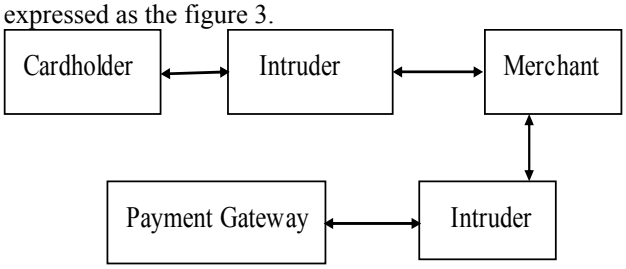

Figure3. The model diagram of system communication

After the above analysis, abstracting the process of the cardholder, its part codes are as shown below:

```
active proctype cardholder()
{
    bit flagC=0;
    mtype No,temp;
    mtype key,key2,g1,g2,g3,g4,g5,g6,g7;
    if                  //choose the communication object
        ::partyC=I;
        ::partyC=M;
    fi;
    c_m! initialize,partnerC,C,Ckey,temp,temp,temp;
    //send initiate request
    c_m? init_ack,C,g1,key,No,g2,g3;->
    //receive the initiate response
    if
        :: (key==Mkey)->No=TNo;
        ::skip;
    fi;
     c_m! purchase,partnerC,C,Ckey,No,OI,PI;
    //send purchase request
    c_m?purchase_ack,C,g4,key2,g5,g6,g7;
    //receive the purchase response
    flagC=1;
 }
```

C. Verify the Properties

When describe the properties of a system with high requirement of security, people generally use the temporal logic which is an expansion of modal logic. It is used in the behavior description of finite state systems widely. In SPIN, we use the linear temporal logic [9] formula LTL to describe property of the model.

The linear temporal logic LTL is composed of the formula, such as Af, where f represents a path formula

and it only allows atomic proposition of state sub-formula. More precisely, path formula LTL can be described as the follows:

(1) If $p \in AP$, then p is a path formula.

(2) If f and g are path formulas, and then $\neg f$, $f \wedge g$, $f \vee g$, G f, X f, Ff, $f \cup g$ and f R g are all path formulas, X,F,G and $\cup$ are tense operators.

In SPIN, LTL operator is described with the ASCII symbols and it includes connection operators of propositional logic, such as && (and), || (or), -> (contain) and logic (!) and three temporal logic operators:

◇ p: Means p is right in some state points,

□p: Means p is right in all state points,

$p \cup q$: Means p is right in the state points before the first state point in which q is right.

The common method of LTL model checking is to convert the properties which are the negative of LTL formula to the auto machine [10] [11], then compute the intersection between this auto machine and the system's auto machine. If the intersection result is null, then it indicates that the system satisfies all the properties, otherwise the counter example will be generated and the corresponding reasons will be gave out [12].

In this paper, we focus on the description of certification among security properties. The so-called property of certification is referring to mutual authentication between the initiator and responder of the communication to ensure that the claimed identity of the participants in the message is consistent with their true identities.

For the cardholders, they need to make sure that the merchants are legal. For the merchants, they need to make sure that the other persons involved are not frauds. In instances, it can be expressed as:[] (<> ((flagC==1) && (flagM==1)) -> ((partyM==C) && (partyC==M))). If the transaction succeeds, then the cardholder and the merchant should be seen as principals of the transaction.

Similarly, the session between a merchant and a payment gateway may also be attacked by an intruder. it can be expressed as: [](<>((flagM==1) && (flagP==1))-> ((PartyM==P) && (PartyP==M))).If the session is completed successfully, then merchant and the payment gateway should be seen as principals of the session.

The following work is to validate the property of authentication. The result is shown in figure 4 and figure 5.

For figure 4, the attack sequence of the intruder is as follows:

C->I:<I,C,Ckey>
I(C)->M:<M,I,Ikey>
M->I(C):<I,M,Mkey>
I->C:<C,I,Ikey>
C->I:<I,C,Ckey,No,OI,PI>
I(C)->M:<M,I,Ikey,No,OI,PI>
M->P:<P,M,Mkey,Ckey,PI,Nm>
P->M:<M,P,Pkey>
M->I(C):<I,M,Mkey>
I->C:<C,I,Ikey>

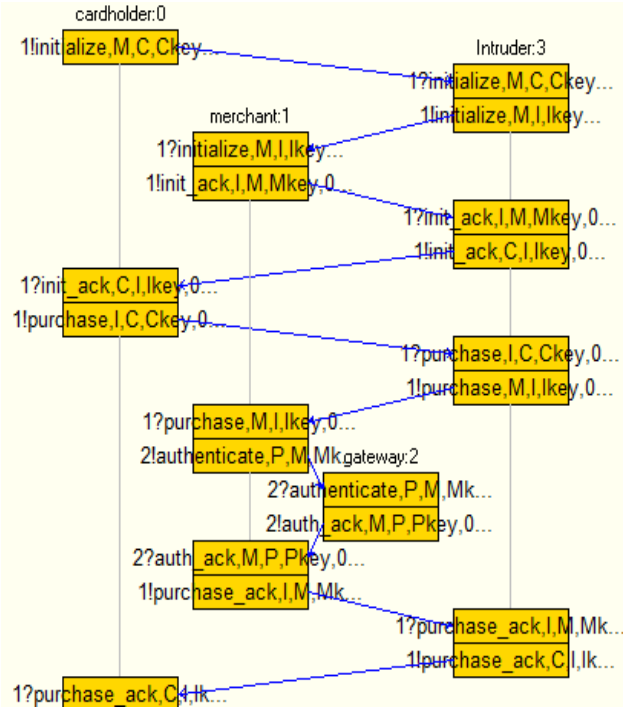

Figure4.Intrude fakes Cardholder to communicate with Merchant

From the attack sequence, we can conclude that the intruder pretends to be the customer and communicate with the merchant, which may result in debiting to the customer's account.

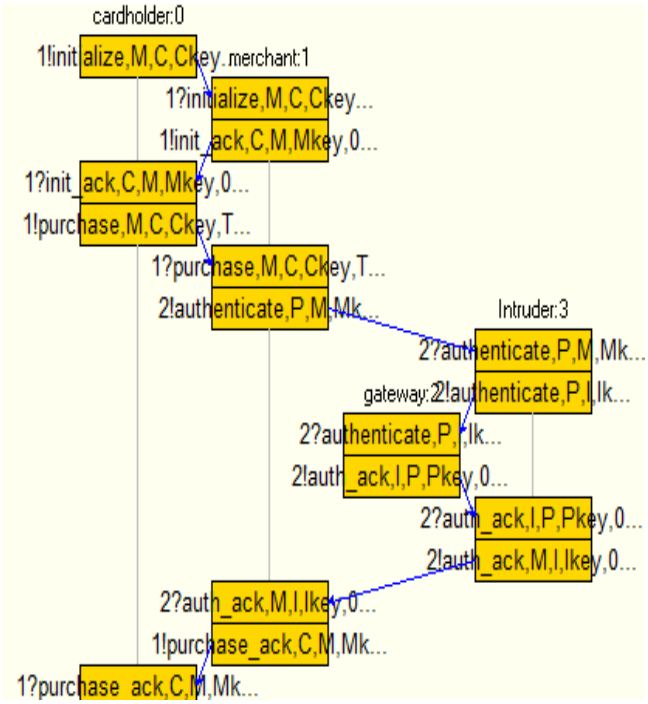

Figure5.Intrude fakes Merchant to communicate with Payment Gateway

For figure 5, the attack sequence of the intruder is as follows:

C->M:<M,C,Ckey>
M->C:<C,M,Mkey>
C->M:<M,C,Ckey,TNo,OI,PI>

M->I:<I,M,Mkey,Ckey ,PI,Nm>
I(M)->P:<P,I,Ikey,Ckey,PI,Nm>
P->I(M):<I,P,Pkey>
I->M:<M,I,Ikey>
M->C:<C,M,Mkey>

From the two figures, we can see that the intruder can pretend to be the cardholder and trade with the merchant in the transaction. Of course, it may also pretend to be the merchant and communicate with the cardholder. Furthermore, in the session of the merchant and the payment gateway, the intruder can also pretend to be the merchant and deceives the payment gateway. Thus, it makes the real merchant and payment gateway cannot communicate normally. The result of the verification is as shown in figure 6.

From figure 6 we can see that the value of errors is not zero, but equals one. This indicates that assertions are violated. The existence of the attack is also confirmed.

```
(Spin Version 6.2.4 -- 8 March 2013)
Warning: Search not completed
         + Partial Order Reduction

Full statespace search for:
         never claim         + (e1)
         assertion violations + (if within scope of claim)
         acceptance   cycles         + (fairness disabled)
         invalid end states   - (disabled by never claim)

State-vector 112 byte, depth reached 41, errors: 1
     18 states, stored
      1 states, matched
     19 transitions (= stored+matched)
      0 atomic steps
hash conflicts:       0 (resolved)

Stats on memory usage (in Megabytes):
   0.002  equivalent memory usage for states (stored*(State-
   0.286  actual memory usage for states
  64.000  memory used for hash table (-w24)
   0.343  memory used for DFS stack (-m10000)
  64.539  total actual memory usage
```

Figure6.The result of verification

D. Model Optimization

The purpose of the model optimization is to improve efficiency of verification. The optimization techniques that we use include two aspects: (1) Reduce the state space that needs to be searched. (2) Reduce the storage space that needs to preserve states [13].

Taking into account the uncertain changes of state space caused by cross execution of concurrent process, we use the atomic sequence of statement in the process of modeling. It represents that the sequence of statement is performed as an indivisible whole, thus, the behaviors of cross execution among the processes can be reduced. If so, the atomization of local computing will play a role in state compression and preventing the occurrence of race conditions. Atomic is a mechanism to implement atomization of local computing. Atomic can make the statements $\{stat_1; stat_2 \ldots stat_n\}$ as an atomic execution sequence. The execution process can't be interrupted by other processes during from the beginning of first statement $stat_1$ to the ending of last statement $stat_n$. When the $stat_i$ is executable, statements of atomic is enabled. When $stat_j$ is blocking, atomic chain is broken but other processes will perform the next step. When the statement becomes executable from blocked, the atomicity recovers and continues execution.

Being different from the general hash table, Bit-state hashing doesn't save each state. It saves a reachable state with a binary bit in the memory. For a given state, it uses the hash function to calculate addresses that save the binary bits in the hash table. Thus, there are no conflict problems of hash. In this paper, we use two kinds of optimization strategies: atomic sequence and Bit-state hashing to compress the state space and storage space in the process of verification, thus, we can reduce the time and space complexity of verification algorithm. The result of the optimized verification is as shown in figure 7and figure 8.

```
     12 states, stored
      1 states, matched
     13 transitions (= stored+matched)
      7 atomic steps
hash conflicts:       0 (resolved)

Stats on memory usage (in Megabytes):
   0.001  equivalent memory usage for states (stored*
   0.285  actual memory usage for states
  64.000  memory used for hash table (-w24)
   0.343  memory used for DFS stack (-m10000)
  64.539  total actual memory usage
```

Figure7. The result of verification using atomic steps

```
     12 states, stored
      1 states, matched
     13 transitions (= stored+matched)
      7 atomic steps

hash factor: 87381.3 (best if > 100.)

bits set per state: 3 (-k3)

Stats on memory usage (in Megabytes):
   0.001  equivalent memory usage for states (stored*
   0.125  memory used for hash array (-w20)
   0.038  memory used for bit stack
   0.343  memory used for DFS stack (-m10000)
   0.664  total actual memory usage
```

Figure8. The result of verification using Bit-state hashing

Table I is the optimization result of the payment process, where No1 represents the data with no use of atomic strategy, No2 represents the data with the use of atomic optimization strategy, No3 represents the data with the use of Bit-state hashing and the atomic optimization strategies. By comparing and analyzing the experimental data, we find that the atomic sequence can reduce the state space effectively and the Bit-state hashing can compress storage space greatly.

TABLE I.
THE OPTIMIZATION OF THE EXPERIMENTAL RESULTS

Optimization strategy	Stored states	State transitions	Peak memory(Mbyte)
No1	18	19	64.539
No2	12	13	64.539
No3	12	13	0.664

IV. The Improvement and Analysis Of Set Protocol

Analyzing the protocol vulnerabilities exploited by these two attacks, we find that there are three major leaks in the first reference.

(1) In the message 3, M can't authenticate the sender's identity of data item Em (OI). The first attack takes advantage of this weakness.
(2) In the message 4, P is unable to authenticate the sender's identity of data item Ep (TransId, AA, MA). The second attack takes advantage of this weakness.
(3) There is no the identity information of the recipient being included in the message 5. So the recipients cannot determine whether the message is sent to them or not, the second attack also takes advantage of this weakness.

To implement the authentication for message sender, we can use the sender's private key to sign the messages, thus, we can prevent the attacker from forging the message [14]. Moreover, the messages contain the identity information of the recipients, which can prevent the attacker forwarding or replaying the messages. For the above vulnerabilities, we improved the protocol as follows:

(1) In the message 3, the data item Em (OI) is signed with the private key of C to achieve the authentication of the message source.
(2) In the message 4, the data item Ep (TransId, AA, MA) is signed with the private key of M to achieve the authentication of the message source.
(3) In the message 5, the recipients' identity information is added into the data item Sp (TransId, TrRes) to prevent the attackers forwarding the message.

The formal description of improved protocol is as follows:

Step 1:C->M: <TransId_Req>
Step 2:M->C: <Sm(TransId)>
Step 3:C->M: <Sc(Em(OI)),Sc(Ep(PI)),DS>
Step4:M->P:
<Sc(Ep(PI)),DS,Sm(Ep(TransId,AA,MA))>
Step 5:P->M: <Sp(TransId,M,C,TrRes)>
Step 6:M->C: <Sp(TransId,M,C,TrRes)>

Where DS represents double signature, it can ensure correlation between the ordering information and payment instruction to prevent the merchant from denying the information of selling goods.

We can prevent above attacks by using the improved protocol and find no new attacks in the analysis of improved protocol. Therefore, the security of the protocol has been further improved.

V. Conclusion

At present, the research and verification of SET protocol using the tool of SPIN is still in infancy. Compared with the logic verification, model checking can be performed automatically. When the model does not meet the system properties, the model checker will give a counter example automatically that doesn't not meet the specifications of system [15]. Therefore, using the tool of SPIN to verify the security of complex SET protocol is an effective method.

In this paper, we use model checking technology to analyze formally the simplified version of SET payment protocol proposed by Lu &Smolka, and find its existing attacks. By analyzing the principle of the attacks, we improve the protocol and avoid the attacks effectively. Meanwhile, we adopt two kinds of strategies to reduce the number of states and implement storage optimization. Thus, we will improve the verification efficiency to some extent.

Acknowledgment

The authors wish to thank the anonymous reviewers for their valuable comments and suggestions to improve the paper.

References

[1] Shiyong Lu,Smolka. Model Checking the Secure Electronic (SET) Protocol. In Proceedings of 7th International Symposium On Modeling, Analysis *and Simulation of Computer and Telecommunication Systems*,pp.358-365,1999.
[2] Simei Lu, Jianlin Zhang. Improvement of SET protocol model and formal analysis via SMV[J].*Computer Engineering and Applications*,vol.8, pp. 113-116,2010.
[3] E.M. Clarke, O. Grumberg,D.A. Peled, Model checking, Cambridge, *MA: MIT Press*, 1999.
[4] Huiling Shi, Wenke Ma, Meihong Yang,Xinchang Zhang. A Case Study of Model Checking Retail Banking System with SPIN[J]. *Journal of computers,* vol. 7(10), pp.2503-2510, 2012.
[5] Cuicui Li, Analysis and verification of E-commerce Protocol Based on SPIN model checking [D].*East China University of Science and Technology*,2012.
[6] Xue Rui,Feng Dengguo. The formal analysis technique and method of security protocol[J].*Journal of Computer*,vol.1,pp.1-20,2006.
[7] Meihua Xiao, Jinyun Xue. Verification of Concurrent Systems Using SPIN/Promela[J]. *Computer Science*, vol.8(31),pp.201-204,2004.
[8] G.J.Holzmann, The SPIN Model Checker, Primer and Reference Manual[M].*Addison-Wesley*,2003
[9] Shenghong Li, Yanqing Feng. Formal analysis of third-party payment based SPIN[J]. *Computer Era*.pp.23-25,2008.
[10] Peled D. Software Reliability Methods [M].*Berlin: Springer2Verlag*, 2001.
[11] CongHua Zhou,Bo Sun. Abstraction In Model Checking Real-Time Temporal Logic of Knowledge[J].*Journal of Computers*,vol.7(2),pp.362-370,2012.
[12] Mingyu Ji, Di Wu, Yanmei Li, Zhiyan Chen. Counterexample Generation for Conditional Probability in Probabilistic Model Checking [J]. *Journal of Computers*, vol.8 (12), pp.3272-3279, 2013.
[13] Chire Wen. Research on SET protocol optimization and

verification of SPIN in small amount of transactions [J]. *Computer and Modernization*, pp.201-206, 2011.
[14] Ruoyan Zhang. Establishment of formal model and security analysis for SET protocol [J].*Computer Applications and Software*, pp.81-84,2009.
[15] Mordechai Ben-Ari. Principles of the SPIN Model Checker, *Springer Verlag*,2008.

Meihua Xiao graduated from Department of Computer Engineering of University of Shanghai for Science and Technology (USST) in 1987. He received the M.S. degree in computer software and theory in 2001 from Nanchang University, and the Ph.D. degree in computer software and theory in 2007 from Institute of Software, Chinese Academy of Sciences, China. From Sept. 2008 to Sept. 2009, he worked as a visiting scholar at Cornell University. He is a Professor at School of Software, East China Jiaotong University, China. His research interests include network and information security, information management and information systems, networks and multimedia teaching.

Zilong Wan received the B.E. degree in computer software from Software College of Nanchang University, Jiangxi, China in June 2011. He is currently a graduate student and working towards his Master of IT at Nanchang University, Jiangxi, China in Jun 2014. His current research interest includes computer software and theory.

Hongling Liu received the B.E. degree in computer science and technology from Jianghuai College of Anhui University in June 2011, Anhui, China. She is currently a graduate student and working towards her Master of IT at Nanchang University, Jiangxi, China in June 2014. Her current research interest includes computer software and theory.