Information Technology Journal 10 (8): 1468-1483, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.1468.1483

Automated Proof of Resistance of Denial of Service Attacks in Remote Internet Voting Protocol with Extended Applied Pi Calculus

Wei Huang and Bo Meng
School of Computer, South-Center University for Nationalities, MinYuan Road #708,
HongShan Section, Wuhan, 430074, Hubei, China

Abstract: Secure remote internet voting protocols play an important role in electronic government. Owning to the huge damage and hard to prevention of denial of service attacks in security protocols, resistance of denial of service attacks occupy a tiny space and is intricate security requirements for remote voting protocols. Meng protocol is one of the most important remote internet voting protocols that claims to satisfy formal definitions of key properties. In this study firstly the review of the formal model of resistance of denial of service attacks in security protocol are introduced. Then extended applied pi calculus, the mechanized proof tool ProVerif and Huang's formal model are examined. After that Meng protocol is modeled in extended applied pi calculus. Finally resistance of denial of service attacks is analyzed with ProVerif. The results we obtain are that Meng protocol is not resistance of denial of service attack because one denial of service attacks is found by us. At the same time we give the method against the denial of service attack. To our best knowledge we are conducting the first mechanized proof of resistance of denial of service attacks in Meng protocol for an unbounded number of honest and corrupted voters.

Key words: Automatic verification, protocol state, symbolic model, Pi Calculus

INTRODUCTION

With the development of Internet and information technology, electronic government has got serious attention from enterprise and academic world. Owning to advantages of remote internet voting, it plays an important role in electronic government. In order to assess its security and increase confidence of the voters in remote internet voting system and protocols, many researchers have pay attention to design and verification on secure remote internet voting systems and protocols. So how to develop and verify a practical secure internet voting protocol is a challenging issue (Meng, 2009c, 2011b).

The practical secure remote internet voting protocol should include privacy, completeness, soundness, fairness and invariableness, universal verifiability, receipt-freeness and coercion-resistance. Previous research focused on implementation and formal analysis of receipt-freeness and coercion-resistance (Meng, 2009c). Besides these properties, owning to the huge damage and hard to prevention of denial of service attacks in security protocol, the secure remote internet voting protocol should also have resistance of denial of service attacks. In the last twenty years many remote internet voting protocols (Clarkson *et al*., 2008; Meng, 2007, 2009a; Meng *et al*., 2010a), claimed on their security, have be proposed. To our best knowledge until now resistance of denial of service attacks in these remote internet voting protocols has not been analyzed.

Owning to the huge damage and hard to prevention of denial of service attacks in security protocol and network, people pay serious attentions on analysis, verification and prevention of denial of service attacks (Tritilanunt, 2009). Denial of service attacks are attacks against availability, attempting to prevent legitimate users from accessing the network and distributed system. This kind of attack aims at rendering a network an system incapable of providing normal service by targeting either the network, bandwidth or connectivity. According to the methods of attacks, denial of service attacks can be classified into three types: resource exhaustion, out of service and physical destroy. Denial of service attacks is simple and effective, for example, the adversary can create many bogus messages and sent to target of attack. That make the target of attack can not provide normal service for legitimate user owning to process big bogus messages. At the same time it is not easy to find the adversary and adversary can mount another type attacks through denial of service attacks, for example, man-in-the-middle attack (Meadows, 2001).

Corresponding Author: Bo Meng, School of Computer, South-Center University for Nationalities, MinYuan Road #708, HongShan Section, Wuhan, 430074, Hubei, China Tel: 0086-18602707481

In order to prevent denial of service attacks the first step is to analyze and proof resistance of denial of service attacks in protocol, network and distributed system with formal method and give the confidence of people in its security. There are two models can be used: one is symbolic model in which cryptographic primitives are ideally abstracted as black boxes, the other is computational model based on complexity and probability theory. Firstly each model formally defines security properties expected from security protocol and then develop methods for strictly proving that given security protocols satisfy these requirements in adversarial environments. Computational model is complicated and is difficult to get the support of mechanized proof tools. In contrast, symbolic model is considerably simpler than the computational model, proofs are therefore also simpler and can sometimes benefit from mechanized proof tools support. For example: SMV (Mei *et al*., 2009), NRL, Casper, Isabelle, Athena, Revere, SPIN, Brutus, ProVerif, Scyther (Meng, 2011b).

In symbolic model there are mainly three formal frameworks in resistance of denial of service attacks. One is Yu-Gligor formal model (Yu and Gligor, 1990) based on user agreement. The core of framework is based on access control policy. It does not deal with denial of service attacks are executed before authentication between sender and receiver in protocol, for example, SYN floods attacks. At the same time framework does not support the automated tools. The other is Meadows's cost-based formal model (Meadows, 2001) which built on the notion that a protocol is a sequence of operations with cause-effect relationships: an action by one principle usually causes a sequence of actions by another principle that incurs some cost. He argues that his formal framework can be supported by modification of automated tools, for example, NRL protocol analyzer. Tritilanunt *et al.* (2007) and Tritilanunt (2009) think that the cost analysis has only taken into account honest runs of the protocol in Meadows's cost-based formal model. At the same time they also think that Meadows used only a coarse measure of computational cost with three levels denoted as cheap, medium or expensive. In practice it can be quite difficult to classify and compare operations in such a coarse measure. The third one is Huang *et al*. formal model (Huang *et al*., 2011) which is the first automatic method of resistance of denial of service attacks based on theorem proof with first order theorem prover ProVerif. ProVerif is a mechanized proof of cryptographic protocol verifier based on a representation of the protocol by Horn clauses or applied pi calculus. It can handle many different cryptographic primitives, including shared- and public-key encryption and signatures, hash functions and Deffie-Hellman key agreements, specified both as rewrite rules and as equations. It can also deal with an unbounded number of sessions of the protocol and an unbounded message space. When ProVerif cannot prove a property, it can reconstruct an attack, that is, an execution trace of the protocol that falsifies the desired property. ProVerif has been tested on protocols of the literature with very great results. In computational model resistance of denial of service attacks analysis model have not been proposed by Blanchet (2001).

Meng protocol (Meng, 2009b) is one of the most important remote internet voting protocols that claims to satisfy formal definitions of key properties without strong physical constrains. Until now resistance of denial of service attacks in Meng protocol is not analyzed. So here we use mechanized proof tool ProVerif to verify resistance of denial of service attacks in Meng protocol based on Huang *et al*. (2011) formal model.

The main contributions of this paper are summarized as follows:

- Review the formal model of resistance of denial of service attacks in security protocol. There are mainly three formal frameworks in resistance of denial of service attacks: Yu-Gligor formal model, Meadows's cost-based formal model and Huang *et al*. (2011) model which is the first automatic method of resistance of denial of service attacks based on theorem proof with first order theorem prover ProVerif. Until now resistance of denial of service attacks analysis model based on computational model have not been proposed
- Apply the mechanized formal model proposed by Huang *et al*. (2011) model for mechanized proof of Meng protocol and its resistance of denial of service attacks. Therefore, Meng protocol is modeled in extended applied pi calculus and resistance of denial of service attacks take into account. The proof itself is performed by mechanized proof tool ProVerif developed by Blanchet
- The result we obtain is that Meng protocol has not resistance of denial of service attacks. One denial of service attack is found by us. At the same time we give the method against the denial of service attack. To our best knowledge we are conducting the first mechanized proof of resistance of denial of service attacks in Meng protocol for an unbounded number of honest and corrupted voters

RELATED WORK

In symbolic model there are mainly three formal frameworks in resistance of denial of service attacks: Yu-Gligor formal model (Yu and Gligor, 1990) based on

user agreement, Meadows's cost-based formal model (Meadows, 2001), Huang *et al*. (2011) formal model (Huang *et al*., 2011) based on theorem proof. In computational model resistance of denial of service attacks analysis model has not been proposed. To our best knowledge until now resistance of denial of service attacks in remote voting protocol is not analyzed.

May be one of the first attempts to formalize the notion of resistance of denial of service attacks was done by Gligor (1983, 1984) with maximum waiting time. Gligor defines availability as the guaranty of a maximum specified waiting time for any operation, even in case of concurrent accesses. If a system is resistance of denial of service attacks then any requesting user will wait no more than maximum waiting time units of time before the service is granted.

Yu and Gligor (1990) propose a formal specification on resistance of denial of service attacks based on temporal logic by introduction of notion of user agreement. The core of framework is based on access control policy. They argue that lack of specifications for these agreements makes it impossible to demonstrate denial of service prevention. If the shared services want to resist denial of services attacks the user must to obey the special constrains. They model availability as how to provide a shared service with a specified maximum waiting time. They argue that denial of service attacks is as the liveness which models the finite waiting time police and safety problem. Their specification model consists of the service specifications and the user-agreement specifications. The service specifications describe all the desired operations and properties that must be provided by the shared service. The user-agreement specifications describe all the desired properties. They use their framework to formalize and analyze denial of services attacks in resource allocator in operating system and Dining philosophers' service. It does not deal with denial of service attacks are executed before authentication between sender and receiver in protocol, for example, SYN floods attacks. At the same time framework does not support the automated tools. Bacic and Kuchta (1991) argue that the core problem of resistance of denial of service attacks is resource allocation. They introduce the notion of a resource allocation monitor that has to have the following three reference monitor characteristics: it is tamper-proof and cannot be prevented from operating and guarantees authorized access to resources under its control. Millen (1993) extended Yu-Gligor formal model by representing the passage of time explicitly. The maximum-waiting-time policy can be expressed as easily as a finite-waiting-time policy. At the same time it can also support other policies of probabilistic nature. Policies and user agreements are expressed without temporal logic. He also proposes a resource allocation model for resistance of denial of service attacks. Waiting time policy consists of a security constraint which says that a subject which asks for a resource has to be provided with it with a maximum waiting time. His formal model catches the effect of realistic system behavior by introducing a state transition model of resource allocation and probabilistic waiting time policies. He also thinks that a denial of service protection base is similar to trusted computing base with strong trust assumptions to guarantee that the constraints on behavior can be reliably enforced. Millen's idea is similar to the one suggested by Abadi and Lamport (1993). Both consider some resource management rules and suggest an approach to model and verify properties including liveness, safety and availability. But Millen' formalism differs significantly which is based on a set-theoretic approach and includes n representation of passage of time explicitly.

Meadows (2001) introduce a very important formal framework of resistance of denial of service attacks based on the costs spending on computation by the principles in security protocols. His formal model based on fail-stop protocol. A fail-stop protocol is one that can provide a certain degree of security against attack and will stop if a message that is interfered with is detected or the verification is failed. The framework is built on the notion that a protocol is a sequence of operations with cause-effect relationships: an action by one principle usually causes a sequence of actions by another principle that incurs some cost. According to Meadows's framework when in a protocol execution at which an attacker can send a message to cause a denial of service attack if the cost of creating the message is small with respect to his resources while the cost to the defender to accept and process the message is relatively more expensive. If this relationship of costs between attacker and defender is not true during a protocol execution, then the protocol is resistance of denial of service attacks. He argues that his formal framework can be supported by modification of automated tools, for example, NRL protocol analyzer. He analyzes the station to station protocol and point out that it is not resistance of denial of service attacks. But Meadows's formal model maybe not practical because the costs of generating a bogus message is small than costs of processing and verifying, so every protocols is not resistance of denial of service attacks. Based on Meadows's cost-based formal model Ramachandran (2002) analyzes JFK protocol and point that JFK protocol is resistance of denial of service attacks with the conditions bogus messages are handled in an appropriate way. Smith *et al*. (2006) also analyze JFK

protocol with Meadows's cost-based formal model. They point that because both of the Diffie-Hellman exponentials (g^r and g^i) can be reused the coordinated attackers can launch the denial of services attacks. At the same time they also think that owning to that availability of IP addresses makes that the cost of revealing an address may be more expensive to an initiator. JFK protocol can be denial of service attacks in the presence of attackers is not willing to reveal a source IP address. But we think these arguments are worth discussing. Lafrance and Mullins (2003) present a method based on admissible interference for finding denial of service attacks in security protocols. Using SPPA and Meadows's cost-based framework, they introduce an information flow property called impassivity which detects whenever an enemy process may cause interference, using its low-cost actions, on high-cost actions of other principals. It is based on the fact that such interference may lead to an attack on the protocol by exploiting this single flaw several times and, thus, causing denial of services through resource exhaustion. Their model is suitable to the model of resource exhaustion denial of service attacks. They point out that 1kp electronic payment protocol is not denial of services attacks. Abadi *et al.* (2007) use the observational equivalence relation to formalize denial of services attacks and find JFK protocol is resistance of denial of services attacks. Tritilanunt *et al.* (2007) and Tritilanunt (2009) firstly point out that the cost analysis has only taken into account honest runs of the protocol in Meadows's cost-based formal model. At the same time they also think that Meadows used only a coarse measure of computational cost with three levels denoted as cheap, medium or expensive. In practice it can be quite difficult to classify and compare operations in such a coarse measure. So they use the colored Petri nets to model the denial of services attacks based on cost-based and time-based model and analyzed the HIP protocol. They find that HIP protocol is not resistance of denial of services attacks in the conditions Type 3 adversary or Type 4 adversary. Zhou *et al.* (2008) propose a model based on strand spaces and 4-way handshakes protocol is analyzed. They find that it is not resistance of denial of services attacks.

Huang *et al.* (2011) present the first automatic method of resistance of denial of service attacks based on theorem proof with first order theorem prover ProVerif. They extend the applied pi calculus from the attacker contexts and process expression and then from the view of protocol state, they propose the first automatic method of resistance of denial of service attacks based on extended applied calculus. At the same time they analyze resistance of denial of service attacks in JFK protocol and IEEE 802.11 i four-way handshake protocol. The results they obtained are that JFK protocol is resistance of denial of service attacks and IEEE 802.11 i four-way handshake protocol is not. The methods to prevent resistance of denial of service attacks in IEEE 802.11 i four-way handshake protocol are also proposed.

Besides the formal model of Yu-Gligor formal model, Meadows's cost-based formal model and Huang *et al.* (2011) formal model, Amoroso (1990) emphasizes the need for specifying a service model in terms of prevent (p, c) policies as predicates concerned subjects, resources and resource consumption operations. Denial of service attacks policies are specified based on predicates that specify conditions, using priorities, under which a subject can deny other authorized subjects access to critical objects. He analyzes resistance of denial of service attacks in system V/MLS with prevent (2, 2).

Based on modal logic and deontic logic, Cuppens and Saurel (1999) propose a formal model to formalize availability policy by the four predicates expression of permissions, prohibitions and obligations of subjects: use-right, disposal-right, realization-right and run-right. They use disposal-right predicate to model waiting time policy and use-right to model use time policy. Their formal framework enables users to express their own required availability properties and to formally check these properties over a given system or protocol by simulating its logical specification. Gabillon and Gallon (2003) resembles the Cuppens and Saurel model. The main difference is that we do not have an explicit waiting time policy. They consider an availability policy as a special case of security policy where the distribution of rights varies with the time. They apply their method to analyze other protocols like the ARINC629 CP or Ethernet. Cuppens *et al.* (2006) use the formal security model called Nomad which combines deontic and temporal logics to specify availability requirements. Each availability requirement expressed in the Nomad model is transformed into a security aspect that can be woven into a program. They mainly concern the denial of service attacks in program. Nomad can transform an insecure program into a secure program.

Agha *et al.* (2005) use probabilistic extension of the Maude term rewriting system called PMAUDE to model denial of services attacks and a sublogic of Continuous Stochastic Logic to describe the rate of success of attack and use the statistical model-checking tool VESTA to analyze the TCP 3 3-way Handshaking protocol and find it is not resistance of denial of services attacks. Mahimkar and Shmatikov (2005) use the alternating time temporal

logic (ATL) to model bandwidth consumption and resource exhaustion attacks and verify JFKr using MOCHA a model checker. They find that JFKr is resistance of denial of services attacks.

REVIEW OF HUANG ET AL. FORMAL MODEL

In this section we firstly review the extended applied pi calculus, then the definitions of resistance of denial of services, finally the method of automated proof of resistance of denial of service attacks.

EXTENDED APPLIED PI CALCULUS

In this section we review extended applied calculus which is based on applied pi calculus. Applied pi calculus (Abadi and Fournet, 2001) is a language for describing concurrent processes and their interactions based on Dolev-Yao model and is an extension of the pi calculus that inherits the constructs for communication and concurrency from the pure pi-calculus. It preserves the constructs for generating statically scoped new names and permits a general systematic development of syntax, operational semantics equivalence and proof techniques. At the same time there are several powerful automatic tool supported applied pi calculus, for example, ProVerif. Applied pi calculus with ProVerif has been used to study a variety of complicated security protocols, such as Just Fast Keying protocol (Abadi *et al*., 2004, 2007), remote electronic voting protocol (Backes *et al*., 2008a; Meng *et al*., 2010c, 2010b, 2011a), a key establishment protocol, direct anonymous attestation protocol (Backes *et al*., 2008b), TLS protocol (Bhargavan *et al*., 2008) .

In order to model the protocol state and resistance of denial of service attacks, Huang *et al*. (2011). extend the applied pi calculus from two aspects: one is the adversary context, the other is process expression. In the following we review the extended applied pi calculus. The extended applied pi calculus is also supported by ProVerif. Here we only review the adversary contexts and process context in extended applied calculus. The other content in extended applied calculus can be found in the reference (Huang *et al*., 2011).

Adversary contexts: In applied pi calculus the adversary is in Dolev-Yao model. But in extended applied pi calculus, according to abilities of adversary the contexts of adversary are classified into two contexts: one is ideal context, the other is real context. Real context is formalized as:

$$\nu\tilde{n}.C\left[C\left[\bar{c}\langle u\rangle\right]\bar{u}\langle N\rangle.P\right],\nu\tilde{n}.C\left[C\left[c(u)\right]u(x).P\right]$$

C ::=	process context
[ ]	null process context
P | C	parallel composition
C | Q	parallel composition
!C	replication
υn.C	name restriction
if M = N then C else Q	conditional
if M = N then P else C	conditional
in(u, x).C	message input
out(u, N).C	message output

Fig. 1: Process context

where, $u\in\tilde{n},c\notin\tilde{n}$ real context is insecure environments where the adversary is in Dolev-Yao model. The adversary in real context can overhear, intercept and synthesize any message and is only limited by the constraints of the cryptographic methods used. Ideal context is formalized as:

$$\nu\tilde{n}.C\left[\bar{u}\langle N\rangle.P\right],\nu\tilde{n}.C\left[u(x).P\right]$$

where $u\in\tilde{n}$. Ideal context is secure environments. The adversary in ideal context can not overhear, intercept and synthesize any message.

Process context: Process context in Fig. 1 is a process with a hole []. The process 0 is an empty process context. The process Q/P is the parallel composition of P and Q. The replication !c produces an infinite number of copies of c which run in parallel. The process un.C firstly creates a new, private name then executes as C. The process in (u, x). C receives a message from channel u and runs the process context c by replacing formal parameter x by the actual message. We use in (u.M). C for the input of terms $M_1,\ldots, M_t$. The process out (u, N).C is firstly ready to output the message N on the channel u and then runs the process context c. The process is the abbreviation for the output of terms $\text{out}\left(u,\tilde{N}\right).C$. The conditional construct if M = N then C else Q runs that if M and N are equal, executes process context C, then C is a verified context. The conditional construct if M = N then P else C runs that if M and N are not equal, executes c, then c is not a verified context.

DEFINITIONS OF RESISTANCE OF DENIAL OF SERVICE ATTACKS

Here, we review related definitions of resistance of denial of service attacks in Huang *et al*. (2011).

Definition 1: An annotated Alice-and-bob specification in protocol: An annotated Alice-and-bob specification in protocol consists of n statements of form:

$$A \to B: R_1^i, \cdots, R_m^i \| M_i \| O_1^i, \cdots, O_k^i$$

where, i∈ [1, n], M_6, denotes the ith message in protocol.

Protocol consists of n messages exchanged between two principles A, B. A statement:

$$A \to B: R_1^i, \cdots, R_m^i \| M_i \| O_1^i, \cdots, O_k^i$$

describes that firstly the sequences of operations $R_1^i, \cdots, R_m^i$ performed by principles A to generate a message M_i and then it is sent to principle B, finally the sequence of operations $O_1^i, \cdots, O_k^i$ performed by principle B. $R_1^i, \cdots, R_m^i$ denotes the sequence of operations performed by principle A for generating M_i. $O_1^i, \cdots, O_k^i$ denotes the sequence of operations performed by principle B after receiving M_i and processing and verifying M_i.

Let:

$$1 = A \to B: R_1^1, \cdots, R_m^1 \| M_1 \| O_1^1, \cdots, O_k^1$$

is an annotated Alice-and-bob specification in protocol, A_{ch} (A) is a set of operations performed by principle A on *1*. $act_1(A)\left[R_1^1, \cdots, R_m^1, M_1\right]$ denotes that the set of the sequence of operations preceding principle $R_1^1, \cdots, R_m^1$ sends message M_i to B.

$$act_1(B)\left[M_1, O_1^1, \cdots, O_k^1\right]$$

denotes that the set of the sequence of operations $O_1^i, \cdots, O_k^i$ performed by principle B after receiving M_1, if any verification operations, for example, decryption, verification of digital signature, failed then the operation stops.

Definition 2: authentication of message M_i: If the statement:

$$1 = A \to B: R_1^1, \cdots, R_m^1 \| M_1 \| O_1^1, \cdots, O_k^1$$

is carry out successfully then the fact that principle B receives message M_i from A is exist; if principle B receives message M_i but principle A does not performance the sequence of operations $act_1(A)\left[R_1^1, \cdots, R_m^1, M_1\right]$, then message M_i received by principle B is altered by the adversary; If message M_i received by principle B is altered by the adversary and B can find the fact that message M_i is modified, then message M_i received by principle B is authenticated.

Definition 3: correspondent in operations: α and β are correspondent in operations if and only if message M_i in:

$$act_i(A)\left[R_1^i, \cdots, R_m^i, M_i\right]$$

are same to message M_i in

$$act_j(B)\left[M_j, O_1^j, \cdots, O_k^j\right]$$

where,

$$i, j \in [1,n],\ \alpha \in act_i(A)\left[R_1^i, \cdots, R_m^i, M_i\right],$$
$$\beta \in act_j(B)\left[M_j, O_1^j, \cdots, O_k^j\right]$$

$$act_i(A)\left[R_1^i, \cdots, R_m^i, M_i\right]$$

denotes that the set of the sequence of operations for generating the message M_i by principle A.

$$act_j(B)\left[M_j, O_1^j, \cdots, O_k^j\right]$$

denotes that the set of the sequence of operations for processing and verifying the message M_i by principle B.

Definition 4: γ_1 casually precedes γ_2: P is an annotated Alice-and-bob specification in protocol, S is a set of all operations in P. For any operation γ_1 and γ_2 in S, γ_1 casually precedes γ_2 if and only if:

If:

$$\gamma_1, \gamma_2 \in act_i(A)\left[R_1^i, \cdots, R_m^i, M_i\right]$$

or:

$$\gamma_1, \gamma_2 \in act_j(B)\left[M_j, O_1^j, \cdots, O_k^j\right], i, j \in [1,n]$$

at the same time γ_1 occurred before γ_2;

If:

$$\gamma_1 \in act_i(A)\left[R_1^i, \cdots, R_m^i, M_i\right],$$
$$\gamma_2 \in act_j(B)\left[M_j, O_1^j, \cdots, O_k^j\right], i, j \in [1,n]$$

at the same time γ_1 and γ_2 are correspondent.

There is operations γ_8, γ_8 casually precedes γ_2, γ_1 casually precedes γ_8.

$act_i(A)\left[R_1^i, \cdots, R_m^i, M_i\right]$ denotes that the set of the sequence of operations for generating the message M_i by principle A. $act_j(B)\left[M_j, O_1^j, \cdots, O_k^j\right]$ denotes that the set of the sequence of operations for processing and verifying the message M_i by principle B.

Definition 5: set of association in message M_i and M_j: Set of association ω between any message M_i and M_j in protocol P is intersection of set Ʊ and set ψ: ω = Ʊ∩ψ, where I, j∈[1, n], i<j, Ʊ is set of data items in verification operations υ in $act_j(B)[M_j, O_1^j, \cdots, O_k^j]$, ψ is the set of data items in message M_i in $act_j(B)[M_j, O_1^j, \cdots, O_k^j]$.

Set of association ω describe the degree and relation of influence among messages in protocol. If ω is null set, then messages in protocol are independent and are not related; if ω is not null and includes many data items, then messages M_i and M_j are related deeply.

Definition 6: resistance of denial of service attacks: P is an annotated Alice-and-bob specification in protocol, B is resistance of denial of service attacks if and only if set of association ω between any message M_i and M_j in set Recv(B):

- ω is null set ∅
- Any data items in ω are authenticated

Where Recv(B) is set where data items are in operations that are ordered in casually precedes in:

$$act_j(B)[M_j, O_1^j, \cdots, O_k^j],\ i, j \in [1,n], i < j$$

If any message M_i and M_j in protocol P are not related, then contexts of processing the message M_i and M_j are independent, then B is resistance of denial of service attacks; if any message M_i and M_j in protocol P are related, then contexts of processing and verifying message M_i and M_j are not independent, then B is resistance of denial of service attacks if and only if set of association ω of any message M_i and M_j in protocol P are authenticated.

METHOD OF AUTOMATED PROOF OF RESISTANCE OF DENIAL OF SERVICE ATTACKS

In this section we review the automated proof of resistance of denial of service attacks in protocol by Hung *et al*. (2011) formal model.

Applying the extended applied pi calculus the protocol can be modeled as an annotated Alice-and-Bob specification. It assumes that the protocol exchanges 2n messages between principles Alice and Bob in a run. Principles Bob receives n messages M_i, i, ∈[1, n]. Principles Bob sends n messages M'_i, i, ∈[1, n]. Protocol process $PP \equiv \nu\tilde{n}.(!Alice|!Bob)$ is a closed process and consists of parallel composition of any initiator processes Alice and responder processes Bob. According to the extended applied pi calculus process Alice and Bob can be reduced into one process in Fig. 2.

$Alice, Bob(\to \cup \equiv)^* 0$	$Alice, Bob(\to \cup \equiv)^* !P$
$Alice, Bob(\to \cup \equiv)^* \nu n.P$	$Alice, Bob(\to \cup \equiv)^* P \mid P'$
$Alice, Bob(\to \cup \equiv)^* c(x).P$	$Alice, Bob(\to \cup \equiv)^* \overline{c}\langle N \rangle.P$
$Alice, Bob(\to \cup \equiv)^*$ if $M = N$ then P else Q	
$Alice, Bob(\to \cup \equiv)^*$ if $M = N$ then P else $C[\overline{c}\langle S \rangle].Q$	$c \notin \tilde{n}$

Fig. 2: Processes

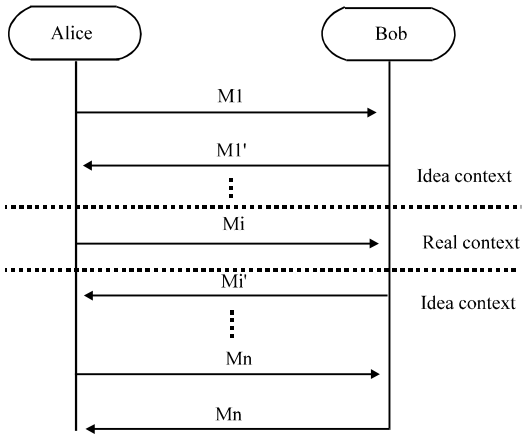

Fig. 3: The formal model of messages M_i i∈ [1, n]

In order to use ProVerif to automatic proof of resistance of denial of service attacks of Bob, the any messages M_i, i, ∈[1, n] is modeled with the extended applied pi calculus. If the adversary can get the secret on the public channel c, then the adversary can launch a denial of service attacks by attacks of message M_i.

The method is used to model the messages M_i, i, ∈[1, n] in Fig. 3. The message M_i is exchanged and processed in real context. The messages

$$M_1, M'_1, \cdots, M_{i-1}, M'_{i-1}, M_i, M_{i+1}, M'_{i+1}, \cdots, M_n, M'_n$$

are exchanged and processed in idea context. Protocol process PP is

$$PP \equiv \nu\tilde{n}.(!Alice_i \mid !Bob_i)$$

c is public channel. c_j, j∈|2, n|∩j≠i are private channels used to receive messages M_i J∈|2, n|∩j≠i.

$$Alice_i(\to \cup \equiv)^* C[\overline{c}\langle c_i \rangle]\overline{c}_i\langle M_i \rangle.Alice_{i+1} \quad c \notin \tilde{n}, c_i \in \tilde{n}$$

$$Bob_i(\to \cup \equiv)^* C[c(x)]x(m_i).Bob_{i+1} \quad c \notin \tilde{n}$$

$$Alice_j(\to \cup \equiv)^* C[\overline{c}_j\langle M_j \rangle]Alice_{j+1} \quad , c_j \in \tilde{n}, j \in [1,n] \cap j \neq i$$

$$Bob_j(\rightarrow \cup \equiv)^* C[c_j(m_j)]Bob_{j+1} \quad , c_j \in \tilde{n}, j \in [1,n] \cap j \neq i$$

If the adversary can get the secret message secret on the public channel c, then the adversary can launch a denial of service attacks by attacks of message M_i.

Theorem: resistance of denial of service attacks: Responder Bob in protocol process PP is resistance of denial of services attacks if and only if the formal model of all messages M_i, i, ϵ[1, n] received by principle Bob, PP can not output the secret message secret, in other words, there is not processes P', P"and attacker process attacker to cause:

$$(PP \mid Attacker)(\rightarrow \cup \equiv)^* \overline{c}\langle S \rangle.P' \mid P'', c \notin \tilde{n}, S \in \tilde{n}$$

According to the theorem, people can use the extended applied pi calculus to model resistance of denial of services attacks in protocol, then based on the proposed theorem, apply ProVerif to automated prove the resistance of denial of services attacks

MECHANIZED PROOF TOOL PROVERIF

ProVerif is an automatic cryptographic protocol verifier based on a representation of the protocol by Horn clauses. It can handle many different cryptographic primitives, including shared- and public-key cryptography (encryption and signatures), hash functions and Deffie-Hellman key agreements, specified both as rewrite rules and as equations. It can also deal with an unbounded number of sessions of the protocol (even in parallel) and an unbounded message space. When ProVerif cannot prove a property, it can reconstruct an attack, that is, an execution trace of the protocol that falsifies the desired property. ProVerif can prove the following properties: secrecy, authentication and more generally correspondence properties, strong secrecy, equivalences between processes that differ only by terms. ProVerif in Fig. 11 has been tested on protocols of the literature with very encouraging results (http://www.proverif.ens.fr/proverif-users.html). Recent research came up with an abstraction of zero-knowledge proofs, a primitive heavily used within electronic voting protocols that are accessible to an automated analysis using ProVerif (Blanchet, 2001).

ProVerif is the only tool for our purpose of an automated verification of Meng protocol based on Huang *et al.* (2011) model. Inspired by works of Huang *et al.* (2011) model we use it to automatically verify resistance of denial of services attacks in Meng protocol.

MENG PROTOCOL

Meng protocol (Meng, 2009b) promises that it can protect voters' privacy and achieves universal verifiability, receipt-freeness and coercion-resistance with weak physical assumptions or procedural constraints. It mainly applies the encryption technologies which include threshold ElGamal cryptosystem, Mix net, homomorphic encryption, Meng non-interactive deniable authentication protocol (Meng, 2009c) and the improved proof protocol that knowledge that two ciphertexts are encryption of the same plaintext. Meng protocol assumes that the private key is private and that the channel between voters and registration authority is one way anonymous channel. Meng protocol consists of the five authorities: registration authority that is responsible for authenticating the voters, issue authority that takes charge of issuing the related key and credentials, bulletin board, voters, tallying authority that is responsible for tallying ballots. The message structure is depicts in Fig. 4.

In preparation phase issuer authority generates the public/private ElGamal keys. The private keys of voter and authorities are secret. It also generates the ballot B^t and send B^t and its digital signature to bulletin board denoted by bulletin board.

In registration phase firstly voter $voter_i$ generates message

$$ENV_{PK_i}(SK_j(ident_j), ident_j, PK_j)$$

and send it to the registration authority A_i. A_i receives the message and open the digital envelope with its private key. He checks $ident_i$ that weather it has registered or not. If it has not registered, he checks Sk_i ($ident_i$).Then A_i generates his public key and public keys for the ciphertext $E^v(c_{i,j})$ of credential shares with the public key of voter, after that he generates $Proof_{V_j}^{A_i}$ based on Meng non-interactive deniable protocol and the improved proof protocol that knowledge that two ciphertexts are encryption of the same plaintext with ElGamal cryptosystem. Finally A_i generates $ENV_{PK_j}(E^v(c_{i,j}), Proof_{V_j}^{A_i})$. Other registration authorities generate

$$ENV_{PK_j}(E^v(c_{i,j}), Proof_{V_j}^{A_i})(i=1,\cdots,i-1,i+1\cdots,s)$$

with the similar method. A_i gets

$$ENV_{PK_j}(E^v(c_{i,j}), Proof_{V_j}^{A_i})(i=1,\cdots,i-1,i+1\cdots,s)$$

from other authorities and sends

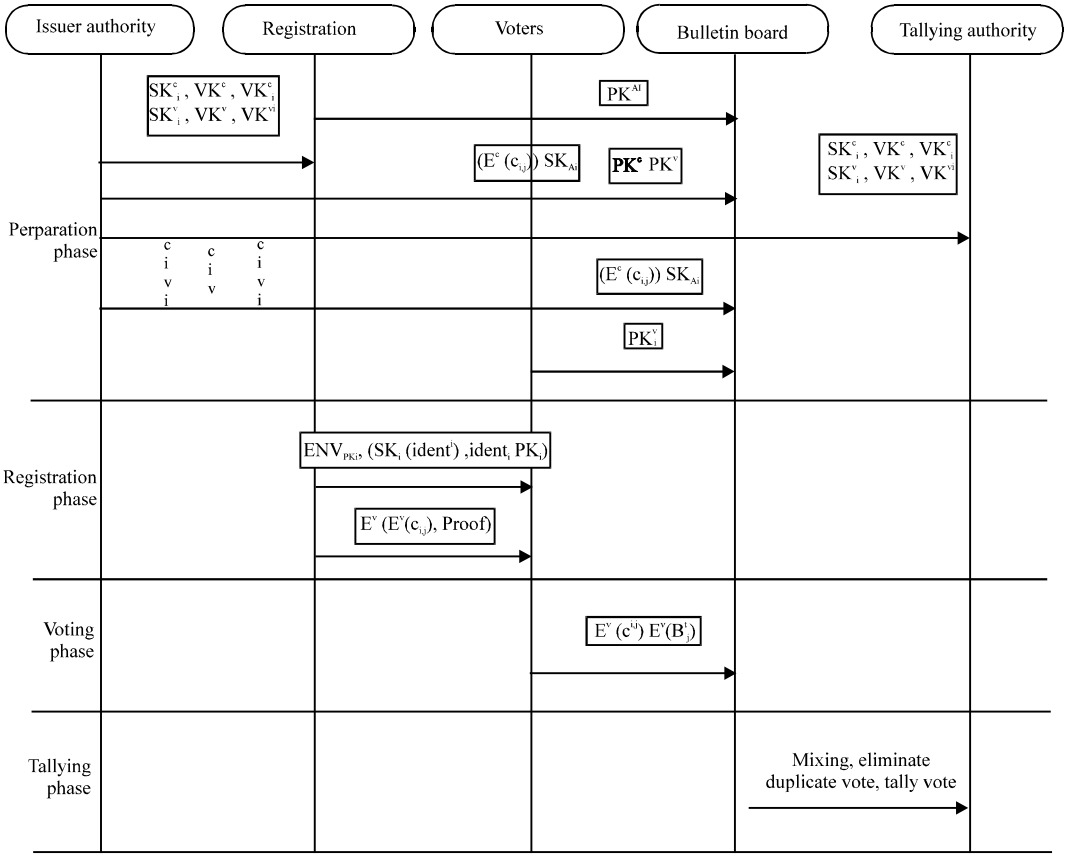

Fig. 4: The structure of message

$$ENV_{PK_j}\left(E^v\left(c_{i,j}\right),Proof_{V_j}^{A_i}\right)(i=1,\cdots,i,\cdots,s)$$

to voter V_i by one way anonymous channel. A_i also generates $(E^o(c_{i,j}))\ Sk_{Ai}$ and sends $(E^o(c_{i,j})$ and $(E^o(c_{i,j}))$ Sk_{Ai} (i = 1,..., i,..., s) to Bulletin Board. After $voter_i$ receives $(E^v(c_{i,j})$ and $(E^o(c_{i,j}))$, he verifies Meng non-interactive deniable protocol proof and proof of equality between $(E^v(c_{i,j})$ and the corresponding $(E^o(c_{i,j})$ that has been signed and published in her reserved area of bulletin board. Upon successful verification, she multiplies together the shares $(E^v(c_{i,j})$ and gets $(E^v(cj)$. Then $voter_i$ chooses his favorite ballot shares $E^v(b_1^t),\cdots,E^v(b_s^t)$ and gets:

$$E^v\left(B_j^t\right)=E^v\left(\sum_{i=1,\cdots,s} b_{i,j}^t\right)$$

then sends $(E^v(cj)$, $E^v\left(B_j^t\right)$ to bulletin board. The registration phase ends.

After the voting time expires, all ballots on bulletin board posted by allegedly eligible voters are mixed by the tallying authorities. Tallying authority publishes the tallying result on bulletin board.

MODELING MENG PROTOCOL WITH EXTENDED APPLIED PI CALCULUS IN HUANG ET AL. FORMAL MODEL

In order to use ProVerif to analyze resistance of denial of service attacks in Meng protocol, in this section we use the extended applied pi calculus to model Meng protocol.

Firstly the function and equational theory is introduced. Cryptography in a Dolev-Yao model is modeled as being perfect. Figure 5 describes the functions and Fig. 6 describes the equational theory in Meng protocol.

Function and equational theory: Encryption algorithm and decryption algorithm in the probabilistic public key cryptosystem is denoted with pPKenc (x, pu, r) and pPKdec (x, PR), respectively. The deterministic public key encryption scheme consists of encryption algorithm

Fun pPKdec(x,PR)	Fun pPKenc(x,PU,r)	Fun SelfBlinding(x,PU)	
Fun PK(x)	Fun SK(x)	Fun sign(x,PR)	Fun add(x,y)
Fun verifysign(x,PU,y)	Fun decsign(x,PU)	Fun NDAMAC(x,PR_S,PU_R)	Fun $projection_i$(x)
Fun TpPKenc(x,PU,r)	Fun TpPKdec(x,PR)	Fun checkciphertext(x_1,x_2)	

Fig. 5: Functions

equation	verifysign(sign(x,SK(y)),PK(y),x)=true.	equation	decsign(sign(x,SK(y)),PK(y))=x.
equation	TpPKdec(SelfBlinding(TpPKenc(x,PK(y),r),PK(y)),SK(y))=x.		
equation	TpPKdec(TpPKenc(x,PK(y),r),SK(y))=x.	equation	add($projection_1$(x),$projection_2$(x))=x.
equation	add($projection_2$(x),$projection_1$(x))=x.	.	
equation	checkciphertext(TpPKenc(x_1,PU_y,r_1),TpPKenc(x_1,PU_z,r_2))=true		
equation	NDAMAC(x,SK(y),PK(z))=NDAMAC(x,PK(y),SK(z))		

Fig. 6: Equational theory

$$\text{Public}_p(ZK_{i,j}(\tilde{N},\tilde{M},F))=N_p. \quad p\in[1,j] \quad \text{Formula}(ZK_{i,j}(\tilde{N},\tilde{M},F))=F.$$
$$\text{Ver}_{i,j}(F,ZK_{i,j}(\tilde{N},\tilde{M},F))=\text{true} \quad \text{iff} \quad \left[F\{\tilde{N}/\tilde{\alpha}\}\{\tilde{M}/\tilde{\beta}\}=\text{true}\right]\wedge\left[F \text{ is an } (i,j)-\text{formula}\right]$$

Fig. 7: Model of zero knowledge proof (Backes *et al*., 2008b)

PK enc (x, PR) and decryption algorithm PK dec (x, PR). Digital signature algorithm includes the generation signature algorithm sugn (x, PR) sign the message x with private key PU and the verification algorithm verify sign (x, PU) verify the digital signature x with public key PU. decsign (x, PU) recover the message from the digital signature x with public key PU. The probabilistic threshold combining algorithm TpPKdec (x_1,..., x_2) recovers x from x_1,..., x . $_2$The deterministic threshold combining algorithm TpPKdec (x_1,..., x_2) means that recover x from x_1,..., x_2. The projection function $projection_i$ (x) generated the ith share from the formatted message x. The self blinding function selfBinding (x, r) blinds message x with r. The add function add (x, y) add x and y. checkciphertext verify the two ciphertext x_1 and x_2 generated with the same plaintext. SK (x), PK (x), VK (x) generated the secret key, public key and verification key of x. NDAMAC (x, PR_S, PU_R) generates MAC in Meng non-interactive deniable protocol.

The equational theory is described in Fig. 6. It also contains and equational rules for abstractly reasoning about the knowledge proof that two ciphertexts are encryption of the same plaintext and used in the voting phase which is modeled as checkciphertext (TpPKenc (x_1, Pu_g, r_1), TpPKenc (x_1, Pu_g, r_2)) = true. It can verify the two ciphertexts TpPKenc (x_1, Pu_g, r_1) generated with the public key PU_y and random number r_2 and TpPKenc (x_1, Pu_g, r_2), TpPKenc (x_1, Pu_g, r_1) generated with the public key PU_z and random number r_1, are the same plaintext x_1. Voter can use the equation NDAMAC (x, SK (y), PK (z)) to compare the MAC receives from registration authority in non-interactive deniable authentication protocol to the MAC generated itself with relative parameters.

Zero knowledge proof modeled in Fig. 7 is used to model the improved proof protocol that knowledge that two ciphertexts are encryption of the same plaintext with ElGamal cryptosystem

Then in the following section we model the process in Meng protocol which consists of the basic process include main process, voter process, corrupted voter process, registration authority process, issuer authority process, BB process and tallying authority process.

The main process in Fig. 8 sets up private channels chVR, $chRI_1$, $chRI_2$ and specifies how the processes are combined in parallel. chVR denotes the private channel between voter and registration authority. $chRI_1$ and $chRI_2$ are the private channel between registration authority and issuer authority, respectively. The main process also generates the key parameters c for credentials, v for vote, s for non-homomorphic cryptosystem, keyV for voter and keyI for issuer authority.

Voter process is modeled in extended applied pi calculus in Fig. 9. Each voter get the non-interactive deniable authentication proof's ciphertexts $kencNDA_1$ and $kencNDA_2$ from registration authority, then decrypt and get the knowledge proofs PEP_1 and PEP_2, the MAC in non-interactive deniable authentication proof $NDAMAC_1$

$$\text{Meng protocol} \triangleq$$
$$\text{new } C;\text{new } V;\text{ new } keyV;\text{new } keyI_1;\text{new } keyI_2;\text{new } keyR;\text{ new } chVR;\text{new } chRI_1;\text{new } chRI_2;$$
$$(!\text{voter}|!\text{corrupted voter}|!\text{tallying authority}|!\text{issuer authority}_1|!\text{issuer authority}_2|!\text{registration authority})$$

Fig. 8: Main process

$$\begin{aligned}
&\text{voter} \triangleq \\
&\quad \text{new}\quad \text{nonce};\ \text{out}(chVR,\text{nonce});\ \text{in}(chVR,id);\text{in}(chVR,kencNDA_1);\ \text{in}(chVR,kencNDA_2); \\
&\quad \text{let}\quad (NDAMAC_1,PEP_1,PK_R)=\text{pPKdec}(kencNDA_1,SK(keyV))\quad \text{in} \\
&\quad \text{let}\quad (NDAMAC_2,PEP_2,=PK_R)=\text{pPKdec}(kencNDA_2,SK(keyV))\quad \text{in} \\
&\quad \text{if}\quad NDAMAC(PEP_1,SK(keyV),PK_R)=NDAMAC_1\quad \text{then} \\
&\quad \Big(\text{if}\quad NDAMAC(PEP_2,SK(keyV),PK_R)=NDAMAC_2\quad \text{then} \\
&\quad\quad \Big(\text{if}\quad \bigwedge_{i=1,2}\text{Ver}_{3,4}(\text{proof}_{PEP},PEP_i)=\text{true}\quad \text{then} \\
&\quad\quad\quad \Big(\text{if}\quad \bigwedge_{i=1,2}\text{checkciphertext}\big(\text{Public}_1(PEP_i),\text{Public}_2(PEP_i)\big)=\text{true}\quad \text{then} \\
&\quad\quad\quad\quad \Big(\text{let}\quad cred=\prod_{i=1,2}\text{Public}_1(PEP_i)\quad \text{in} \\
&\quad\quad\quad\quad\ \text{new}\quad r; \\
&\quad\quad\quad\quad\ \text{let}\quad vote=\text{ballot}^t\in\{v_1\cdots v_i\}\quad \text{in} \\
&\quad\quad\quad\quad\ \text{let}\quad vencvote=\text{TpPKenc}(vote,PK(V),r)\quad \text{in} \\
&\quad\quad\quad\quad\ \text{out}\big(pub,(cred,vencvote)\big).\Big)\quad \text{else}\quad \text{out}(c,secret)\Big)\quad \text{else out}(c,secret)\Big)\quad \text{else out}(c,secret)\Big)
\end{aligned}$$

Fig. 9: Voter process

$$\begin{aligned}
&\text{corrupted voter} \triangleq \\
&\quad \text{new}\quad \text{nonce};\ \text{out}(chVR,\text{nonce});\ \text{in}(chVR,id); \\
&\quad \text{in}(chVR,kencNDA_1);\ \text{in}(chVR,kencNDA_2); \\
&\quad \text{let}\quad (NDAMAC_1,PEP_1,PK_R)=\text{pPKdec}(kencNDA_1,SK(keyV))\quad \text{in} \\
&\quad \text{let}\quad (NDAMAC_2,PEP_2,=PK_R)=\text{pPKdec}(kencNDA_2,SK(keyV))\quad \text{in} \\
&\quad \text{if}\quad NDAMAC(PEP_1,SK(keyV),PK_R)=NDAMAC_1\quad \text{then} \\
&\quad \text{if}\quad NDAMAC(PEP_2,SK(keyV),PK_R)=NDAMAC_2\quad \text{then} \\
&\quad \text{if}\quad \bigwedge_{i=1,2}\text{Ver}_{3,4}(\text{proof}_{PEP},PEP_i)=\text{true}\quad \text{then} \\
&\quad \text{if}\quad \bigwedge_{i=1,2}\text{checkciphertext}\big(\text{Public}_1(PEP_i),\text{Public}_2(PEP_i)\big)=\text{true}\quad \text{then} \\
&\quad \text{let}\quad cred=\prod_{i=1,2}\text{Public}_1(PEP_i)\quad \text{in} \\
&\quad \text{out}(pub,cred).
\end{aligned}$$

Fig. 10: Corrupted voter process

and $NDAMAC_2$, respectively. After that the voter checks $NDAMAC_1$ and $NDAMAC_2$, then uses checkciphertext to verify the equivalence between $public_1$ (PEP_i) and $public_2$ (PEP_i). If the verification is fail then it output secret by public channel c. The voter multiplies:

$$cred=\prod_{i=1,2}\text{Public}_1(PEP_i)$$

Because of the homomorphic properties of ElGamal cryptosystem, the resulting ciphertext cred includes the sum of credential shares. The resulting ciphertext (cred, vencvote = TpPKenc (vote, PK (V), r)) is sent to bulletin board.

Corrupted voters process is modeled in Fig. 10. He can sends cred through a public channel, thus the attacker can get the credential cred, so that the attacker can use the cred to impersonate them in order to mount any sort of attack.

Registration authority process is modeled in Fig. 11 that firstly generates the voters id, then gets the secret credentials shares $cred_1$ and $cred_2$ from issuer authority. Then he creates the ciphertexts of non-interactive deniable authentication proof that the proof of the

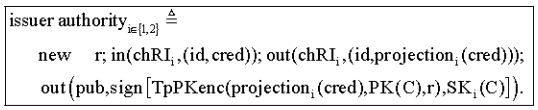

Fig. 11: Registration authority process

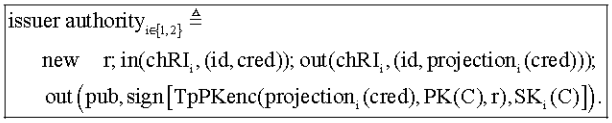

Fig. 12: Issuer authority process

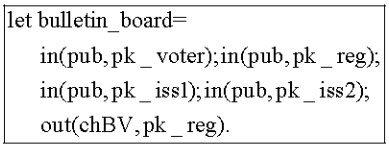

Fig. 13: BB process

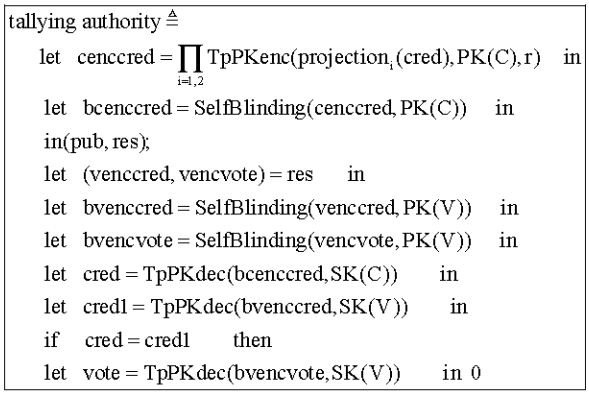

Fig. 14: Tallying authority process

equivalence between the encrypted share sent to the voter pPKenc (NDAMAC (PEP_1, SK (key R), PK (key V)), PEP_1, PK (key R), PK (key V), r) and pPKenc (NDAMAC (PEP_2, SK (key R), PK (key V)), PEP_2, PK (key R), PK (key V), r).

Issuer authority is modeled in Fig. 12 that firstly gets the shares of credential by $projection_i$ (cred), then send sign [TpPKenc ($projection_i$), PK (C), r), Sk_i(C)] which encrypted with a set of ElGamal public parameters by the public channel pub.

BB process is modeled in Fig. 13. BB process receives k-voter, pk-reg, pk-iss1, pk-iss2 through public channel pub and out public key pk-reg from public channel chBV.

Tallying authority process is modeled in Fig. 14. After the voting time expires, the tallying authorities get the all ballots on bulletin board and then mixed it by elfBlinding (cencered, PK (C)). At the same time the shares of credentials posted by the registration authorities are also combined and then mixed selfBlinding (venccredvote, PK (V)). Thus he obtain two lists: a list:

$$\prod_{i=1,2} \text{TpPKenc}(\text{projection}_i(\text{cred}),\text{PK}(C),r)$$

and a set res. The two lists have been encrypted with different ElGamal public parameters. Using threshold protocols for the corresponding sets of private keys, the tallying authorities decrypt the elements in each list by TpPKdec (bcenccred, SK (C)) and TpPKdec (bcenccred, SK (V)) then compare them through a search algorithm and publish the tallying result on bulletin board.

MECHANIZED PROOF OF MENG PROTOCOL WITH PROVERIF

ProVerif can take two formats as input: one is Horn clauses, the other is process in an extension of the pi calculus (Abadi and Blanchet, 2005; Huang *et al*., 2011). In both cases, the output of the system is essentially the same. In this paper we use the extended pi calculus as the input of ProVerif.

In order to prove resistance of denial of services attacks in Meng protocol the formal model in extended applied pi calculus are needed to be translated into the syntax of ProVerif and generated the ProVerif inputs in the extended pi calculus. The code in analysis of resistance of denial of service attacks in voter in Meng protocol is presented in Fig. 15.

The result of resistance of denial of services attacks in Meng protocol in Fig. 16. We find that Meng protocol is not resistance of denial of services attacks because ProVerif out the message "Secret" by public channel c. In Meng protocol there is one resistance of denial of services attack by us: in preparation phase issuer authority publishes public keys Pk^{o}, PK^{v} for voter and his public key PK^{Ai} on BB without protecting security of these public keys by public channels. Thus the adversary can intercept public keys Pk^{o}, PK^{v}, PK^{Ai} and modify it,

```
fun pPKenc/3.  (*probabilistic public key encryption*)
fun pPKdec/2.  (*probabilistic public key decryption*)
fun sign/2.    (*generation signature algorithm*)
fun decsign/2. (*verification signature algorithm*)
fun TpPKenc/3.
fun TpPKdec/2.  (*threshold probabilistic public key decryption*)
fun SK/1.       (*generate the private key*)
fun PK/1.       (*generate public key*)

(*two ciphertexts are encryption of the same plaintext*)
fun checkciphertext/2.
fun add/2.             (*add operation*)
fun multi/2.           (*multi operation*)
fun equals/2.          (*equals test*)
fun selfblinding/2.    (*selfblinding*)
fun projection1/1.     (*projection*)
fun projection2/1.     (*projection*)
fun zk/2.
fun zkver/1.
fun public1/1.
fun NDAMAC/3.

data true/0.

equation pPKdec(pPKenc(x,PK(y),z),SK(y))=x.
equation decsign(sign(x,SK(y)),PK(y))=x.
equation equals(x,x)=true.
equation add(projection1(x),projection2(x))=x.
equation add(projection2(x),projection1(x))=x.
equation multi(TpPKenc(a,PK(y),r),TpPKenc(b,PK(y),z))
=TpPKenc(add(a,b),PK(y),r).
equation TpPKdec(selfblinding(TpPKenc(x,PK(y),r),PK(y)),SK(y))=x.
equation TpPKdec(TpPKenc(x,PK(y),r),SK(y))=x.
equation checkciphertext(TpPKenc(x,PK(y),r1),TpPKenc(x,PK(z),r2))=true.
equation public1(zk(x,y))=y.
equation zkver(zk((cred,r1,r2),(TpPKenc(cred,PK(V),r1),
TpPKenc(cred,PK(C),r2))))=true.
equation equals(NDAMAC(x,SK(y),PK(z)),NDAMAC(x,SK(z),PK(y)))=true.

(*public channel*)
free pub,pubR,pubI1,pubI2.
private free chvote.
free va,vb.
free n1,n2.
private free Secret.

                                        new r;
                                        in(chvote,vote);
                                        let vencvote=TpPKenc(vote,PK(V),r) in
                                        out(pub,(cred,vencvote))
                                        )else out(pub,Secret)
                                        )else out(pub,Secret)
                                        )else out(pub,Secret)
                                        )else out(pub,Secret).
                                    let corruptedvoter=in(chBV,(=n1,nonceB));
                                        new keyV;
                                        out(pub,(n2,nonceB,PK(keyV)));
                                        new nonce;
                                        out(chVR,(n1,nonce,PK(keyV)));
                                        in(chVR,(=n2,=nonce,kencNDAMAC1,kencNDAMAC2));
                                        let (NDAMAC1,PEP1)=pPKdec(kencNDAMAC1,SK(keyV)) in
                                        let (NDAMAC2,PEP2)=pPKdec(kencNDAMAC2,SK(keyV)) in
                                        in(chBV,PK_R);
                                        if equals(NDAMAC(PEP1,SK(keyV),PK_R),NDAMAC1)=true  then
                                        if equals(NDAMAC(PEP2,SK(keyV),PK_R),NDAMAC2)=true then
                                        if zkver(PEP1)=true then
                                        if zkver(PEP2)=true then
                                        let (vencered1,cencered1)=public1(PEP1) in
                                        let (vencered2,cencered2)=public1(PEP2) in
                                        if checkciphertext(vencered1,cencered1)=true then
                                        if checkciphertext(vencered2,cencered2)=true then
                                        let cred=multi(vencered1,vencered2)  in
                                        out(pub,cred).vote,SK(V)) in  0.

                                    let registration_authority=
                                        in(chRT,(=n1,nonceT));
                                        in(chVR,(=n1,nonceV,pk_voter));
                                        new cred;
                                        new nonce1;
                                        new nonce2;
                                        out(chRI1,(n1,nonce1,cred));
                                        out(chRI2,(n1,nonce2,cred));
                                        in(chRI1,(=n2,=nonce1,cred1));
                                        in(chRI2,(=n2,=nonce2,cred2));
                                        new r;new r1;new r2;
                                        let PEP1=zk((cred1,r1,r2),(TpPKenc(cred1,PK(V),r1),TpPKenc(cred1,PK(C),r2))) in
                                        let PEP2=zk((cred2,r1,r2),(TpPKenc(cred2,PK(V),r1),TpPKenc(cred2,PK(C),r2))) in
                                        out(chVR,(n2,nonceV,pPKenc((NDAMAC(PEP1,SK(keyR),pk_voter),PEP1),pk_voter,r),
                                    pPKenc((NDAMAC(PEP2,SK(keyR),pk_voter),PEP2),pk_voter,r)));
                                        out(chRT,(n2,nonceT,TpPKenc(cred1,PK(C),r1),TpPKenc(cred2,PK(C),r2))).
```

Fig. 15: Continued

```
query attacker:Secret.
let votechooser =
  out(chvote,va) | out(chvote,vb).

let voter= in(chBV,(=n1,nonceB));
  new keyV;
  out(pub,(n2,nonceB,PK(keyV)));
  new nonce;
  out(chVR,(n1,nonce,PK(keyV)));
  in(chVR,(=n2,=nonce,kencNDAMAC1,kencNDAMAC2));
  let (NDAMAC1,PEP1)=pPKdec(kencNDAMAC1,SK(keyV)) in
  let (NDAMAC2,PEP2)=pPKdec(kencNDAMAC2,SK(keyV)) in
  in(chBV,PK_R);
  if equals(NDAMAC(PEP1,SK(keyV),PK_R),NDAMAC1)=true  then
  (
  if equals(NDAMAC(PEP2,SK(keyV),PK_R),NDAMAC2)=true then
  (
  if zkver(PEP1)=true then
  (
  if zkver(PEP2)=true then
  (
  let (venccred1,cenccred1)=public1(PEP1) in
  let (venccred2,cenccred2)=public1(PEP2) in
  if checkciphertext(venccred1,cenccred1)=true then
  if checkciphertext(venccred2,cenccred2)=true then
  let cred=multi(venccred1,venccred2)  in

 let issuer_authority1=
    in(chRI1,(=n1,nonceR,cred));
    let cred1=projection1(cred) in
    out(chRI1,(n2,nonceR,cred1));
    new r;
    out(pub,(sign(TpPKenc(cred1,PK(C),r),SK(keyI1)),TpPKenc(cred1,PK(C),r))).

let issuer_authority2=
   in(chRI2,(=n1,nonceR,cred));
   let cred2=projection2(cred) in
   out(chRI2,(n2,nonceR,cred2));
   new r;
   out(pub,(sign(TpPKenc(cred2,PK(C),r),SK(keyI2)),TpPKenc(cred2,PK(C),r))).

let bulletin_board=new nonceV;
    out(chBV,(n1,nonceV));
    in(pub,(=n2,=nonceV,pk_voter));
    in(pubR,pk_reg);
    in(pubI1,pk_iss1);
    in(pubI2,pk_iss2);
    out(chBV,pk_reg).
process  new C;new V;
   new keyR;  new keyI1;  new keyI2;  new chVR;  new chRI1;  new chRI2;  new chRT;
   new chBV;  new chBR;  out(pub,PK(C));  out(pub,PK(V));  out(pubR,PK(keyR));
   out(pubI1,PK(keyI1));  out(pubI2,PK(keyI2));
   ((!voter)|(!corruptedvoter)|(!tallying_authority)|(!registration_authority)|
(!issuer_authority1)|(!issuer_authority2)|(!votechooser )|(!bulletin_board))
```

Fig. 15: The code in analysis of resistance of denial of service attacks in voter in Meng protocol

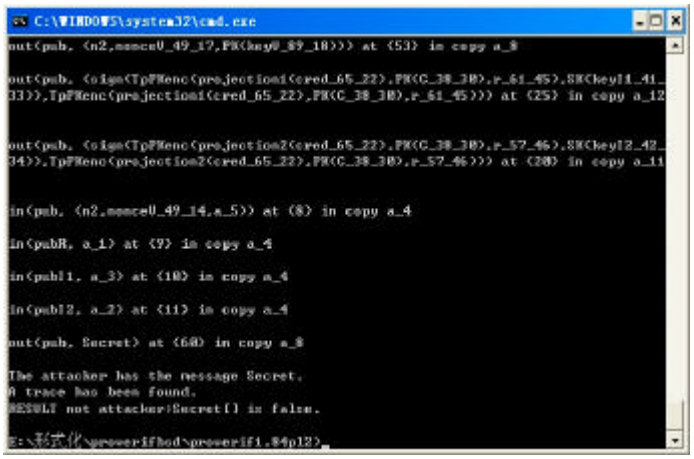

Fig. 16: The result of resistance of denial of services attacks in Meng protocol

then send it to BB. In voting phrase voter v_i firstly verifies non-interactive deniable authentication proof $DAMAC_1$ and $DAMAC_2$, then he uses cheekciphertext to check the equivalence between the encrypted share $public_1$ (PEP) and the one E^o (c), the voter has received to its message is also provided to itself. PK^{Ai} has been publish on BB with digital signature with authority. Owning the adversary has modified the public keys PK^{Ai}, hence the verification of $DAMAC_1$ and $DAMAC_2$ failed, thus voter v_i can not vote. Hence make a resistance of denial of services attack. In order to protect Meng protocol against the denial of service attack we can use the digital certificate to distribute these public keys: Pk^o, PK^v and PK^{Ai} .

ACKNOWLEDGMENT

This study was supported in part by Natural Science Foundation of The state Ethnic Affairs Commission of PRC under the grants No: 10ZN09, titled "Research on the Provably Secure Remote Internet Voting Protocols without Physical Constrains", conducted in Wuhan, China from 1/1/2011 to 30/12/2011.

CONCLUSION

Internet voting protocol play an important role in remote voting system. Owning to the huge damage and hard to prevention of denial of service attacks in security protocol, the secure remote internet voting protocol should have resistance of denial of service attacks.

Recently Huang *et al.* (2011) proposed an automatic model that can be used to analyze the denial of service attacks in security protocol, so Meng protocol can be proved with mechanized proof tool ProVerif. In this paper the review formal model of resistance of denial of service attacks in security protocol are presented. Then apply the mechanized formal model proposed by Huang *et al*. (2011) The result is that Meng protocol has not resistance of denial of service attacks. One denial of service attack is found by us. At the same time we give the method against the denial of service attack.

As future work we plan to prove other resistance of denial of service attacks internet voting protocols. It would also be interesting to formalize the security properties in wireless communication protocol in the formal model with mechanized proof tool ProVerif. At the same time we will formalize the security properties of remote internet voting protocols in the computational model with mechanized tool CryptoVerif.

REFERENCES

Abadi, M. and L. Lamport, 1993. Composing specifications. ACM Trans. Prog. Lang. Syst., 15: 73-132.

Abadi, M. and C. Fournet, 2001. Mobile values, new names and secure communication. Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, London, UK., March 2001, ACM New York, USA., pp: 104-115.

Abadi, M., B. Blanchet and C. Fournet, 2004. Just fast keying in the Pi calculus. Proceeding of the 13th European Symposium on Programming Languages and Systems: Barcelona, Spain, March 29-April 2, Springer Berlin/Heidelberg, pp: 340-354.

Abadi, M. and B. Blanchet, 2005. Analyzing security protocols with secrecy types and logic programs. J. ACM, 52: 102-146.

Abadi, M., B. Blanchet and C. Fournet, 2007. Just fast keying in the Pi calculus. ACM Trans. Inform. Syst. Sec., 10: 1-59.

Agha, G., M. Greenwald, C.A. Gunter, S.K.J. Meseguer, K. Sen and P. Thati, 2005. Formal modeling and analysis of DoS using probabilistic rewrite theories. Proceedings of International Workshop on Foundations of Computer Security. Chicago IL, 2005. http://www.cis.upenn.edu/~mbgreen/papers/fcsw05.pdf.

Amoroso, E., 1990. A policy model for denial of service. Proceedings of Third IEEE Computer Security Foundations Workshop, June 12-14, Franconia, New Hampshire, USA., pp: 110-114.

Bacic, E. and M. Kuchta, 1991. Considerations in the preparation of a set of availability criteria. Proccedings of 3rd Annual Canadian Computer Security Symposium (ACCSS'91), Ottawa, Canada, pp: 283-292.

Backes, M., C. Hritcu and M. Maffei, 2008a. Automated verification of remote electronic voting protocols in the applied Pi-calculus. Proceedings of the 21st IEEE Computer Security Foundations Symposium, June 23-25, EEE Computer Society, Washington, DC, pp: 195-209.

Backes, M., M. Maffei and D. Unruh, 2008b. Zero-knowledge in the applied pi-calculus and automated verification of the direct anonymous attestation protocol. Proceedings of the 29th IEEE Symposium on Security and Privacy, May 2008, Preprint on IACR ePrint, pp: 202-215.

Bhargavan, K., R. Corin, C. Fournet and E. Zalinescu, 2008. Cryptographically verified implementations for TLS. Proceedings of the 15th ACM Conference on Computer and Communications Security, Oct. 27-31, Alexandria, Virginia, USA., pp: 459-468.

Blanchet, B., 2001. An efficient cryptographic protocol verifier based on prolog rules. Proceedings of the 14th IEEE Workshop on Computer Security Foundations, June 11-13, IEEE Computer Society, Washington, DC., pp: 82-96.

Clarkson, M.R., S. Chong and A.C. Myers, 2008. Civitas: Toward a secure voting system. Proceeding of the 2008 IEEE Symposium on Security and Privacy, May 18-21, Oakland, California, USA., pp: 354-368.

Cuppens, F. and C. Saurel, 1999. Towards a formalization of availability and denial of service. In Information Systems Technology Panel Symposium on Protecting Nato Information Systems in the 21st Century, Washington, http://www.irisa.fr/lande/jensen/Dispo/Author/CUPPENS-F.html.

Cuppens, F., N. Cuppens-Boulahia and T. Ramard, 2006. Availability enforcement by obligations and aspects identification. Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06), IEEE Computer Society, Washington, DC, USA, pp: 229-239.

Gabillon, A. and L. Gallon, 2003. An availability model for avionic data buses. Proceedings of WISP03. http://www.univ-pau.fr/~gallon/publis/wisp03.pdf.

Gligor, V.D., 1983. A Note on the denial-of-service problem. Proccedings of IEEE Symposium on Security and Privacy, Apr. 25- 27, Oakland, CA, pp: 139-139.

Gligor, V.D., 1984. A note on denial-of-service in operating systems. IEEE Transactions on Software Engineering, May 29, IEEE Computer Society, pp: 320-324.

Huang, W., B. Meng and D.J. Wang, 2011. Automatic proof resistance of denial of service attacks in protocols. J. Commun.,

Lafrance, S. and J. Mullins, 2003. Using admissible interference to detect denial of service vulnerabilities. Proceedings of the 6th International Workshop in Formal Methods, July 11, Dublin City University, pp: 1-17.

Mahimkar, A. and V. Shmatikov, 2005. Game-based analysis of denial-of-service prevention protocols. Proceedings of the 18th IEEE Workshop on Computer Security Foundations, June 20-22, Aix-en-Provence, pp: 287-301.

Meadows, C., 2001. A cost-based framework for analysis of denial of service networks. J. Comput. Security, 9: 143-164.

Mei, J., H. Miao and P. Liu, 2009. Applying SMV for security protocol verification. Infom. Technol. J., 8: 1065-1070.

Meng, B., 2007. An internet voting protocol with receipt-free and coercion-resistant. Proceedings of 7th IEEE International Conference on Computer and Information Technology, Oct. 16-19, IEEE Computer Society, Washington DC, USA., pp: 721-726.

Meng, B., 2009a. A critical review of receipt-freeness and coercion-resistance. Inform. Technol. J., 8: 934-964.

Meng, B., 2009b. A secure internet voting protocol based on non-interactive deniable authentication protocol and proof protocol that two ciphertexts are encryption of the same plaintext. J. Networks, 4: 370-377.

Meng, B., 2009c. A secure non-interactive deniable authentication protocol with strong deniability based on discrete logarithm problem and its application on Internet voting protocol. Inform. Technol. J., 8: 302-309.

Meng, B., W. Huang and J. Qin, 2010a. Automatic verification of security properties of remote internet voting protocol in symbolic model. Inform. Technol. J., 9: 1521-1556.

Meng, B., W. Huang, Z. Li and D. Wang, 2010b. Automatic verification of security properties in remote internet voting protocol with applied pi calculus. Int. J. Digital Content Technol. Appl., 4: 88-107.

Meng, B., Z. Li and J. Qin, 2010c. A receipt-free coercion-resistant remote internet voting protocol without physical assumptions through deniable encryption and trapdoor commitment scheme. J. Software, 5: 942-949.

Meng, B., 2011a. A survey on analysis of selected cryptographic primitives and security protocols in symbolic model and computational model. Inform. Technol. J., 10: 1068-1091.

Meng, B., 2011b. Refinement of mechanized proof of security properties of remote internet voting protocol in applied PI calculus with proverif. Inform. Technol. J., 10: 293-334.

Millen, J.K., 1993. A resource allocation model for denial of service protection. J. Comput. Security, 2: 89-106.

Ramachandran, V., 2002. Analyzing DoS-resistance of protocols using a cost-based framework. Technical Report DCS/TR-1239, Yale University, USA., http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.132.3873.

Smith, J., J.M. Gonzalez-Nieto and C. Boyd, 2006. Modelling denial of service attacks on JFK with Meadows's cost-based framework. Proc. Aust. Workshops Grid Comput. e-Res., 54: 125-134.

Tritilanunt, S., 2009. Protocol engineering for protection against denial of service attacks. Doctor thesis, Queensland University of Technology, Brisbane Australia.

Tritilanunt, S., C. Boyd, E. Foo and J.M.G. Nieto, 2007. Cost-based and time-based analysis of DoS-resistance in HIP. Proc. Aust. Conf. Comput. Sci., 62: 191-200.

Yu, C.F. and V.D. Gligor, 1990. A formal specification and verification method for the prevention of denial of service. IEEE Trans. Software Eng., 16: 581-592.

Zhou, S.J., R. Jing and X.H. Yang, 2008. DoS attacks on security protocols of the formal analysis. J. Res. Instit. China Electron., 3: 592-598.