http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 11 (7): 741-750, 2012
ISSN 1812-5638 / DOI: 10.3923/itj.2012.741.750

Automatic Formal Framework of Coercion-resistance in Internet Voting Protocols with CryptoVerif in Computational Model

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

Abstract: Automatic proof for internet voting protocols is a hotspot issue in security protocol world. To our best knowledge, until now analysis of coercion-resistance and internet voting protocols with automatic tool in computational model does not exist. So in this study, we initiatively proposed the automatic framework of coercion-resistance and internet voting protocols based on computational model with active adversary. In the proposed framework observational equivalence is used to formalize coercion-resistance. At the same time we propose a method to prove the observational equivalence between the two processes have the same structure and differ only by the terms and term evaluations with mechanized tool CryptoVerif. Based on the proposed method to prove with CryptoVerif the observational equivalence, the proposed mechanized framework can be used to automatically analyze coercion-resistance of internet voting protocols with CryptoVerif.

Key words: Computational model, automatic verification, observational equivalence, coercion-resistance

INTRODUCTION

With the development of information technology, many nations have implemented internet voting system. Securities of internet voting protocols have attracted researchers' attention and effort to implement and prove it. Until know many voting protocols have been proposed and claims on their securities (Juels and Jakobsson, 2002; Juels *et al*., 2005; Acquisti, 2004; Meng, 2009a; Araujo *et al*., 2008; Meng and Wang, 2010; Meng *et al*., 2010a). Generally the secure internet voting protocol should have basic properties and expanded properties including receipt-freeness (Benaloh and Tuinstra, 1994) and coercion-resistance (Juels *et al*., 2005). Especially receipt-freeness and coercion-resistance are the key properties in internet voting protocols (Meng, 2009c).

In order to verify security properties of security protocols including remote internet voting protocols and increase the public's confidence, two distinct frameworks including symbolic framework (Derakhshandeh *et al*., 2008; Hashemi *et al*., 2012; Samimi and Golkar, 2012) and computational framework have been developed for analyzing security protocols form the beginning of the 1980s.

Cryptographic primitives are modeled as black boxes in symbolic framework. So, it is simple than computational framework and is easy to get the help of the automatic tools, for example, SMV, NRL, Casper, Isabelle, Athena, Revere, SPIN, Brutus, ProVerif, Scyther and AVISPA. Owning its ideal abstraction of cryptographic primitives, hence the result is unrealistic. Computational framework is based on complexity and probability. In computational framework the attacker is modeled as a probabilistic polynomial-time Turing machine and a protocol is an unbounded number of copies of probabilistic polynomial-time Turing machine. If an adversary can win an attack game with non-negligible probability, then a predefined security is invalid. Hence the results of proof are clear and practical, while is also complexes and highly error prone. The computation framework is more realistic, but it is difficult to automatic proof until the introduction of mechanized tool CryptoVerif (Blanchet, 2008) which is the only automatic tool with computational framework to our knowledge.

In symbolic framework, Delaune *et al*. (2006a) have done a pioneering work on the formal definition of receipt-freeness and coercion-resistance based on applied pi calculus. But Jonker and De Vink (2006) point out that the formal model (Delaune *et al*., 2006a) offers little help to identify receipts when receipts are present. Hence they present a new formal method using the process algebra to analyze receipts. About the model Meng (2009d) argues that it is worth discussing. Hence he gives a formal logic framework for receipt-freeness based on V. Kessler and H. Neumann logic and find that the voting protocol (Fujioka *et al*., 1992) is not receipt-freeness. Backes *et al*. (2008) initiatively propose an automatic model of coercion-resistance, receipt-freeness and soundness in remote internet voting protocols based on applied

pi calculus with ProVerif and the analysis indicates that the voting protocol (Juels *et al*., 2005) is coercion-resistance with some conditions.

To our best knowledge, until now there is no analysis of coercion-resistance and internet voting protocols with automatic tool in computational model. Hence analysis of coercion-resistance and internet voting protocols with automatic tool in computational model is a significant work. In this study, we use observational equivalence in extend Blanchet calculus to model and automatically prove coercion-resistance of internet voting protocols with mechanized tool CryptoVerif.

CONTRIBUTION AND OVERVIEW

During the past few decades internet voting protocols has been studied deeply. A lot of internet voting protocols have been proposed which claims that have the security properties, for example, receipt-freeness and coercion-resistance and so on.

There are two frameworks which can be used to increase the confidence of people in internet voting protocols. One is symbolic framework (Sharma *et al*., 2007; Nabi *et al*., 2010; Darmawan *et al*., 2009). The other is computational framework that base issues of complexity and probability. The computation framework is more realistic. To our best knowledge, until now there have not exist analysis of security properties and internet voting protocols with automatic tool in computational model.

So analysis of security properties and internet voting protocols with automatic tool in computational framework a hotspot issue in security protocol world and is a significant work. The main contributions of this paper are summarized as follows in detail:

- The state-of-art of proof in symbolic framework and in computational framework is presented. We find that security properties and internet voting protocols are analyzed with informal method, or with symbolic framework. Until now there have not exist analysis of internet voting protocols and its security properties with automatic tool in computational framework
- Review CryptoVerif and propose a method to prove with CryptoVerif the observational equivalence between the two processes have the same structure and differ only by the terms and term evaluations containing in computational framework
- The observational equivalence and private channels in extended Blanchet calculus are used to model coercion-resistance and internet voting protocols, after that we propose the first mechanized framework of coercion-resistance in internet voting protocols in computational framework with active adversary. The

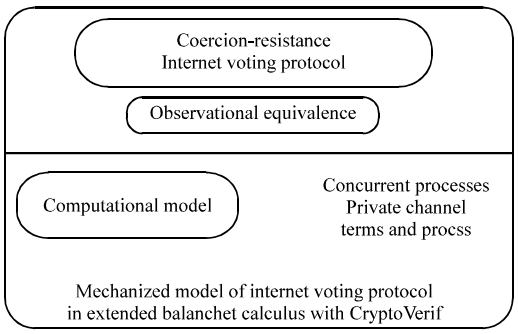

Fig. 1: Analysis model of internet voting protocols with extended Blanchet calculus

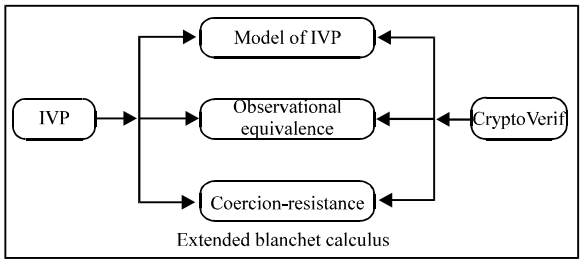

Fig. 2: The idea of automatic framework of coercion-resistance

coercion-resistance is expressed by observational equivalence. Based on the proposed method to prove with CryptoVerif the observational equivalence, the mechanized framework can be used to automatically analyze coercion-resistance of internet voting protocols with CryptoVerif. Fig. 1 describes the analysis model of coercion-resistance in internet voting protocols with extended Blanchet calculus and CryptoVerif

- In order to give the model of coercion-resistance in computational model, based on extended Blanchet Calculus we first model the Internet Voting Protocol (IVP) and then the model of coercion-resistance is presented. Fig. 2 describes the idea of automatic framework of coercion-resistance

Related work: In order to verify the security properties of security protocols and increase the public's confidence, two frameworks have been developed for analyzing security protocols form the beginning of the 1980s. Since 1980's two distinct frameworks: symbolic framework (Mousa, 2005; Rabah, 2005; Zaidan *et al*., 2010) and computational framework are proposed to verify the security properties of security protocols and cryptographic primitives.

In the following section we mainly survey the symbolic proof and computational proof on coercion-

resistance. We find that security properties and internet voting protocols are analyzed with informal method, or with symbolic framework. Until now there does not exist analysis of internet voting protocols and its security properties with automatic tool in computational model.

Delaune *et al.* (2006a) have done a pioneering work on the formal definition of receipt-freeness and coercion-resistance based on applied pi calculus. They use adaptive simulation to formalize coercion-resistance. The ideas is that whenever the coercer requests a given vote on the left-hand side then V_B can change his vote according to the right-hand side and counterbalance the outcome. The voting protocol (Lee *et al.*, 2003) is analyzed with their formal model. Meng (2008) also applies their formal model to analyze the voting protocol (Meng, 2007a). Delaune *et al.* (2006b) also use applied pi calculus to model fairness, eligibility, privacy, receipt-freeness and coercion-resistance and analyze the voting protocols (Fujioka *et al.*, 1992; Lee *et al.*, 2003). But Jonker and De Vink (2006) point out that the formal model (Delaune *et al.*, 2006a) offers little help to identify receipts when receipts are present. Hence they present a new formal method using the process algebra to analyze receipts. Meng (2007b) analyzes receipt-freeness of the several voting protocols based on their formalism.

To our knowledge all previous formal models on coercion-resistance are not getting the help of the automatic tools. Backes *et al.* (2008) propose the first formal model of receipt-freeness and coercion-resistance in remote internet voting protocol in applied pi calculus with ProVeif. In order to formalize coercion-resistance, the process called Extractor is introduced. Extractor extracts the vote the coercer casts on behalf of V_i and tallies it directly. We argue that Extractor is so very powerful that is not practical. Based on Backes *et al.* (2008) model, analysis in Meng *et al.* (2010c) indicates Meng *et al.* (2010b) has coercion-resistance while is not soundness because ProVerif found an attack on soundness. Then the improvement of Meng *et al.* (2010b) is proposed; at the same time analysis in Meng (2011) points out that Acquisti (2004) has soundness and coercion-resistance in some conditions; analysis in Meng *et al.* (2010b) shows that Meng (2009d) protocol has coercion-resistance while has not soundness because ProVerif found an attack on soundness, then the improvement of Meng (2009d) protocol is proposed.

To our best knowledge, until now there does not exist analysis of coercion-resistance and internet voting protocols with automatic tool in computational model. So, in this study we use extended Blanchet calculus to automatically analyze coercion-resistance in the internet voting protocols with CryptoVerif.

REVIEW OF EXTENDED BLANCHET CALCULUS

Here, we review extended Blanchet calculus. In extended Blanchet calculus, messages are bitstrings and cryptographic primitives are functions operating on bitstrings. Extended Blanchet calculus is adapted from the pi calculus and its semantics is purely probabilistic. All processes run in polynomial time: Polynomial number of copies of processes and length of messages on channels bounded by polynomials. Extended Blanchet calculus consists of terms and processes and can get the help of mechanized proof CryptoVerif. The semantic of extended Blanchet calculus is same to the Blanchet calculus and also can get the help of CryptoVerif.

Extend output process In extended Blanchet calculus, the output process new χ $[i_1,.....i_m]$: T;P chooses a new random number uniformly in I_η, (T) stores it in χ $[I_1,...i_m]$ and executes P. Function symbols represent deterministic functions, so all random numbers must be chosen by new χ $[i_1,...,i_m]$: T. Deterministic functions make automatic syntactic manipulations easier: it can be duplicated by a term without changing its value. The process Let χ $[i_1,.....,i_m]$: T = M in P stores the bitstring value of M in χ $[i_1,....,i_m]$ and executes P. The conditional construct if defined $(M_1,...M_i)/\backslash$ M then P else P runs that if Defined $(M_1,...,M_i)/\backslash M$ is true, executes P, otherwise executes P in real context. The conditional construct if defined $(M_1,...,M_i)/\backslash M$ then P else C [existdos (M,N)] ; P^{1}runs that if defined $(M_1,....,M_i)$ is true, executes P, otherwise executes C [existdos (M,N)] .P in idea context. Find $(\oplus^m_{j=1}U_{J1},....,U_{jmj}$ [I] $\leq n_{jmj}$ such that defined $(M_{j1},...M_{jlj})$ $/\backslash$ M_j then P_j) else P means that it tries to find a branch J in [1,M] such that there are values of $u_{j1},...,$ u_{jm} for which $M_{j1},...M_{jlj}$ are defined and M_j is true. In case of success, it executes P. In case of failure for all branches; it executes P. The formula event e $(M_1,...M_m)$ holds when the event e $(M_1,...,M^m)$ has been executed. This event does not change the state of the system and just record that a certain program point has been reached, with certain values of the arguments of the event.

Model of private channel: The method of model of private channel in extended Blanchet calculus is reviewed which can also get the help of CryptoVerif. The method of model of private channel is that find operation is used to simulate a private channel owning to that find operation is get the information in a way but the adversary does not access the information. For example in Fig. 3 in the following protocol, if principle A want to send nonce to principle B in secret channel: A--->B: nonce; principle B get the nonce and commutates f (nonce): B--->A: f (nonce). We can use find operation to model a private channel.

MECHANIZED PROOF TOOL CRYPTOVERIF

Here, we give a brief overview of the mechanized prover CryptoVerif. In most cases, it succeeds in proving the desired properties when they hold, and obviously it always fails to prove them when they do not hold. In other words CryptoVerif is sound but not complete which means that it cannot prove are not necessarily invalid.

CryptoVerif can directly prove security properties of cryptographic protocols in the computational model in which the cryptographic primitives are functions on bit-strings and the adversary is a polynomial-time Turing machine. It also can proves secrecy properties and events that can be executed only with negligible probability, also it can handles various cryptographic primitives, for example, MACs, stream and block ciphers, public-key encryption, signatures, hash functions. CryptoVerif works for N sessions with an active adversary. In a recent case study, CryptoVerif is used to verify: FDH signature scheme, PKINIT for Kerberos, Protocol Implementations in ML, the basic and public-key Kerberos protocol, Protocol Implementations for TLS, Diffie-hellman protocol, denial authentication protocol and electronic payment protocol.

Games approach: CryptoVerif prove security using the sequence-of-games approach. Figure 4 describes the idea of security properties proof with sequences of games:

- Define the desired security properties of given security protocol and cryptographic primitives and attack game0 which is the original attack game with respect to a given efficient adversary and cryptographic primitive initialized
- Generally makes transformation between successive games based on one of the methods: Indistinguishability, failure events and bridging steps and then generate new attack game

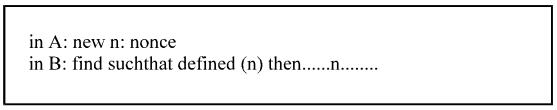

Fig. 3: A single session

- Evaluate the change between two consecutive attack games
- Check desired security properties in attack game, if the change between original game and final game is very small that it can be negligible, the proof is completed and the desired security properties are proved

Produced proofs: In CryptoVerif the process calculus represents games and proofs are represented as sequences of games in Fig. 5, where the initial game formalizes the protocol for which one wants to prove certain security properties. In a proof sequence, two consecutive games $Q^1$and Q^2 are observationally equivalent, meaning that they are computationally indistinguishable for the adversary. CryptoVerif transforms one game into another by applying the security definition of a cryptographic primitive or by applying syntactic transformations. In the last game of a proof sequence the desired security properties should be obvious. Each transformation between two consecutive games preserves polynomial-time Turing indistinguishability.

- Use Blanchet calculus to formalize the investigated protocol and desired security properties are obvious in initial game generated by CryptoVerif which is the real protocol, in interaction with an adversary
- Apply transformations, which are rewriting rules that yield a game either equivalent or almost equivalent under a computational assumption, include the security definition of a cryptographic primitive and syntactic transformations; the game makes transformation and generates new game. Between consecutive games, the difference of probability of success of an attack is negligible or bounded
- Compute the probability of distinguishing two consecutive games based on observational equivalence
- Executes Simplify and tests whether the desired security properties are proved at the beginning of the proof and after each successful cryptographic transformation. The last game is ideal: The security

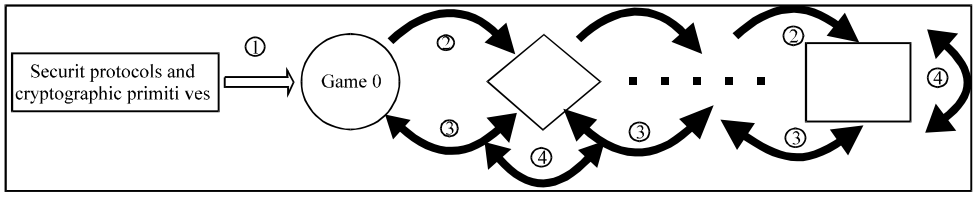

Fig. 4: The ideal of security properties proof with sequences of games

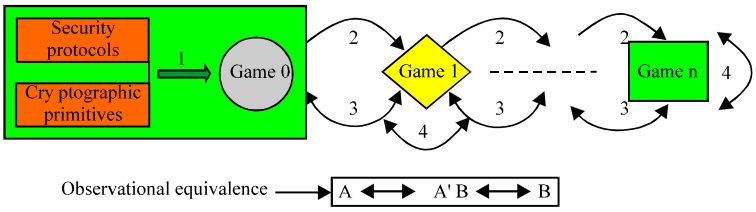

Fig. 5: The ideal of security properties automatic proof in CryptoVerif

property is obvious from the form of the game. The advantage of the adversary is typically 0 for this game.

Proof technique: In CryptoVerif the proof technique is called game transformations which can make the game transformations which allow it to transform the process that represents the initial protocol into a process on which the desired security property can be proved directly. It transforms a game 0 into an observationally equivalent game using:

- Security definition of a cryptographic primitive which is defined by observational equivalences L≈R given as axioms and that come from security properties of primitives. These equivalences are used inside a context: Game1 ≈ Context [L] ≈ Context [R]≈ Game2

Syntactic transformations: Simplification, expansion of assignments, we obtain a sequence of games Game0≈ Game1≈.....≈GameM, which implies Game0≈GameM.

If some equivalence or trace property holds with overwhelming probability in Game1, then it also holds with overwhelming probability in Game0.

Proof strategy: At the beginning of the proof and after each successful cryptographic transformation, CryptoVerif executes Simplify and tests whether the desired security properties are proved. If so, it stops. In order to perform the cryptographic transformations and the other syntactic transformations, our proof strategy relies of the idea of Advice.

Advice means that CryptoVerif tries to apply all equivalences given as axioms, which represent security assumptions. It transforms the left-hand side into the right-hand side of the equivalence. If such a transformation succeeds, the obtained game is then simplified. When these transformations fail, they may return syntactic transformations to apply in order to make them succeed, called advised transformations. CryptoVerif then applies the advised transformations and retries the initial transformation.

Input and output of cryptoverif: The input script in CryptoVerif can be seen as an initial game, modeling the protocol, to which CryptoVerif applies transformations, until a final game that satisfies target security conditions is reached. This proof technique is known as game hopping. CryptoVerif takes as input a script, written in a variant of the pi calculus with an explicit polynomial bound for every replicated process. Thus, processes represent polynomial-time Turing machines that exchange finite bitstrings through an adversary, modeled as a polynomial-time Turing machine. In the script, cryptographic assumptions are introduced through type and function declarations, equations, inequations and game-based equivalences. The equations and inequations are typically used to describe minimal positive assumptions, whilst the game-based equivalences are used to state minimal negative assumptions.

Generally the input consists of the three following parts:

λ: The security assumptions on the cryptographic primitives
λ: The initial game, given in a syntax close to the standard notations in cryptography
λ: The properties to prove

Generally CryptoVerif outputs which consists of the three following parts:

λ: The (negligible) probability that each desired property is wrong
λ: The sequence of games that leads to the proof
λ: A succinct explanation of the transformations performed between games

CryptoVerif works in two modes: A fully automatic and an interactive mode. The interactive mode, which is best suited for protocols using asymmetric cryptographic primitives, requires a CryptoVerif user to input commands that indicate the main game transformations the tool should perform.

AUTOMATED PROOF OF OBSERVATIONAL EQUIVALENCE IN CRYPTOVERIF

Observational equivalence in Fig. 6 in extended Blanchet calculus describes the relationship between two processes. Let review the definition of observational

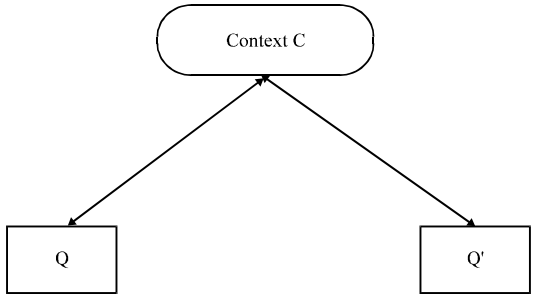

Fig. 6: Observational equivalence

equivalence: Let Q and Q^1 be two processes and V a set of variables. Assume that Q and Q^1satisfy Invariants 1, 2 and 3 and the variables of V are defined in Q and Q^1, with the same types. An evaluation context is said to be acceptable for Q, Q^1, V if and only if var (c) ∩ (VAR (Q)∪ var (Q')⊆V satisfies Invariants 1, 2 and 3. We say that Q and Q are observationally equivalent with public variables V, written $Q\approx^{v}Q'$, when for all evaluation contexts C acceptable for Q,Q', V, for all channels C and bitstrings α, $|P_r[C[Q]\rightarrow \eta C<\alpha>]-P_r[C[Q]\rightarrow \eta C<\alpha>]|$ is negligible. Intuitively, the goal of the adversary represented by context C is to distinguish Q from Q.

We know that CryptoVerif does not directly support the observational equivalence, but we can use the following method to prove observational equivalence between the two processes have the same structure and differ only by the terms and term evaluations in computational model with CryptoVerif:

- Choose a random bit bat the beginning of the process: new b:bool
- If the condition the choice one of the term term1 or term2 in the process, then it is replaced with clauseif be then term1 else term2, then CryptoVerif will expand the check, thus it can distinguish the cases b = true or b = false in the rest of the process
- Prove the secrecy of b. If the result is true then the two processes are observational equivalence

FORMAL FRAMEWORK OF REMOTE INTERNET VOTING PROTOCOLS IN COMPUTATIONAL MODEL

In order to formalize the coercion-resistance, firstly we give the formal model of remote internet voting protocol. The remote voting protocol main process consists of trusted election authority processes an unbounded number of voters' processes and attacker processes. These processes are running in parallel. The voters fall into two categories: Honest voters and coerced voters. Honest voters are issued by registration authority and behave according to the protocol specification. Coerced voters fall into three categories: one is that will registries and abandons. The other is that registries and then simply output all their registration secrets including valid credential identity on a public channel according to the requirement of the attacker and abandons. The third is that registries and then output the fake registration secrets including fake credential identity on a public channel according to the requirement of the attacker and then abandons. Thus that the attacker can impersonate them in order to mount any sort of attack. We assume that the attacker do according to the protocol specification using the fake or valid credential identity of coerced voters in remote voting protocol main process. In our model, we also assume that the channels between authorities and voters in registration phrase are tapple and the channels between voters and bulletin board in the voting phrase are not secure. At the same time the channels among the authorities are secure. We use observational equivalences between voters in remote voting protocol main processes to formalize coercion-resistance. In the following we give the formal definition of remote voting protocol main process, followed by some additional explanations.

DEFINITION: (REMOTE VOTING PROTOCOL MAIN PROCESS)

A remote voting protocol main process is a process Main $P\equiv^{!i\le n}$ c[i](!$^{!i\le n}$Voter^{honest}|!$^{!i\le n}$Voter ^{coerced}|!$i^{\le n}$a Voterta^{tacker} |Reg_{Auth}|$Tally_{Auth}$|...|$Bulletin_{Auth}$) Honest voter process:

$$Voter^{honest} \equiv Cred[i](Cred_{id}:Tcred);$$
$$Voter;new\ vote_{cred}:\widetilde{vote};\overline{Cred[i]}\langle\ \rangle$$

where, $\widetilde{vote}$ is the set of valid votes, Cred [I] is a private channel, Vorer is a process.

Registration authority process:

$$Reg_{cred} \equiv input(\ );newChannel\ RCred[i];Reg;$$
$$new\ RCred_{id}:Trcred;\overline{RCred[i]}\langle RCred_{id}\rangle$$

where, R Cred [i] is a private channel, R $Cred_{id}$ is a variable, Voter is a process.

Coerced voter process $Voter^{coeced}$ fall into three categories $Voter_{valid}^{coecred}$, $Voter_{valid}^{coereod}$ and $Voter_{abs}^{coereod}$. $Voter_{fake}^{Coereod}$. $Voter_{abs}^{coereod};Voter_{abs}^{coereod,}$;≡HCred [i] ($Hcred_{id}$: Thcrd):

$$Voter_{vaild}^{coereod} \equiv HCred[i](HCred_{id}:Thcred);\overline{HvaildCred[i]}$$
$$\langle HCred_{id}\rangle;\overline{CcredStatus[i]}\langle credStatus:Tstatus\rangle$$

$$\text{Voter}_{\text{fake}}^{\text{coerecd}} \equiv \text{HCred}[i](\text{HCred}_{id} : \text{Thcred}); \text{new HfakeCred}_{id} : \text{Thcred};$$
$$\overline{\text{HfakeCred}[i]}\langle \text{HfakeCred}_{id} \rangle; \overline{\text{CcredStatus}[i]}\langle \text{credStatus} : \text{Tstatus} \rangle$$

where, $HCred_{id}$, $HvalidCred_{id}$ and $HfakeCred_{id}$is a variables, $Hcred^{id}$, HValidcred [i] and $HfakeCred_{id}$ are channels, Ctallyvote [i] is a private channel, Cred status is used to indicate the voter of $Cred_{id}$ whether is a coerced voter or not.

Attacker process Voter $^{\text{attacker}}$ consists of two kinds of voter processes $\text{Voter}_{\text{fake}}^{\text{attacker}}$and $\text{voter}_{\text{valid}}^{\text{attacker}}$:

$$\text{Voter}_{\text{fake}}^{\text{attacker}} \equiv \text{CCred}[i](\text{fakeCCred}_{id} : \text{Tccred}); \text{Voter};$$
$$\text{new some} : \text{specialvote}; \overline{\text{CCred}[i]}\langle \text{some} \rangle$$

$$\text{Voter}_{\text{vailed}}^{\text{attacker}} \equiv \text{CCred}[i](\text{vaildCCred}_{id} : \text{Tccred}); \text{Voter};$$
$$\text{new some} : \text{specialvote}; \overline{\text{CCred}[i]}\langle \text{some} \rangle$$

where, Some; is the special vote required by the coercer, CCred[i], is a public channel, $FakeCCred_{id}$ and $validCCred_{id}$ are variables, specialvote is the set of votes, VOTER is a process.

Tallying process:

$$\text{Tally}_{\text{Auth}} \equiv \text{context}\left[\begin{array}{l} \text{CcredStatus}[i](\text{credStatus} : \text{Tstatus}); \\ \text{Tallying}; \overline{\text{Ctallyvote}[i]}\langle \text{ballot} : \text{Tballot} \rangle \end{array}\right]$$

where, Ctallyvote [i] is a private channel, credstatus and ballot are variables, Context is a context, Tallying is a process, Credstatus is used to indicate the voter of $Cred_{id}$ whether is a coerced voter or not.

Bulletin board process:

$$\text{Bulletin}_{\text{Auth}} \equiv \text{Bcontext}\left[\begin{array}{l} \text{CBresu}[i](\text{ballot} : \text{Tballot}); \\ \text{BB}; \overline{\text{CBresu}[i]}\langle \text{result} : \text{Tballot} \rangle \end{array}\right]$$

where, Cbresu [i] is a public channel, result is a variable, BB is a process.

Honest voter process is denoted by $\text{Voter}^{\text{honest}}$. It first receives a credential identity $Cred_{id}$ by a private channel Cred [i] from registration authority, and then it casts his vote $Vote_{cred}$. The credential identity $Cred_{id}$ is generated by registration process Reg_{cred}.

Registration authority process is denoted by Reg_{cred}. It first sets up a private channel Rcred [i] and then executes the process Reg, finally it generates the credential identity $RCred_{id}$ for the voter process Voter and outputs it by the private channel Rcred [i]. The registration process Reg_{cred} is a credential issuer that assigns to each voter a credential that uniquely associates with voter. These credential identities are private and hence does not known to attackers. They are used to make voter processes unique and vote a valid ballot and each voter holding a genius credential identity will be considered eligible in the election. The credential identities that link between a ballot and the voter have to be hidden. If the attacker knows the valid the credential identities then he can use it to impersonate the voter to vote his favor ballot.

Coerced voter process $\text{Voter}_{\text{abs}}^{\text{coerecd}}$ consists of three kinds of voter processes $\text{Voter}_{\text{abs}}^{\text{coerecd}}$, $\text{Voter}_{\text{valid}}^{\text{coerecd}}$ and $\text{Voter}_{\text{fake}}^{\text{coerecd}}$. The coerced voter process $\text{Voter}_{\text{abs}}^{\text{coerecd}}$ means that it receives a credential identity $Hcred_{id}$ from registration authority process and then abandons. The coerced voter process $\text{Voter}_{\text{valid}}^{\text{coerecd}}$ means that it receives a credential identity $HCred_{id}$ from registration authority process then sends it to the attacker and then sends Credstatus to tallying authority process $Tally_{Auth}$, finally it abandons. The coerced voter process $\text{Voter}_{\text{fake}}^{\text{coerecd}}$ means it receives a credential identity $HCred_{id}$ from the registration authority process then he generates a fake credential identity $H_{fake}Cred_{id}$ and sends it to attacker and sends credstatus to tallying authority process $Tally_{Auth}$, finally it then abandons.

Attacker process $\text{Votter}_{\text{attacker}}$ can be clarified in to two categories: one category is denoted by $\text{Voter}_{\text{fakeattacker}}$. It first receives the fake credential identity fake $CCred_{id}$ from the coerced voter process $\text{Voter}_{\text{fake}}^{\text{coerecd}}$ and then votes a special ballot some according to the requirements of the attacker. The other is denoted by $\text{Voter}_{\text{valid}}^{\text{attacker}}$. It first receives the valid credential identity $validCCred_{id}$ from the coerced voter process $\text{Voter}_{\text{valid}}^{\text{coerecd}}$ and then votes a special ballot some according to the requirements of the attacker.

In addition, there also exist tallying authority process $Tally_{Auth}$ that is in charge of tallying the valid votes and outputting them to bulletin board process $Bulletien_{Auth}$ by a private channel Ctallyvote [i]. When it tallying the ballot, it uses the variables credStatus to find the ballots voted by the attacker, then those ballots are eliminated, thus it does affect the soundness of result of tallying. Finally these outputs constitute the result of the election. At the same time bulletin board process $bulletin_{Auth}$ accepts the inputs from voter process and tallying authority process and then publishes the result by a public channel Cbresu [i]. Finally, we define an election context Election Context as a process with a hole that is in parallel composition with the other voter processes and attacker processes.

FORMAL DEFINITIONS OF COERCION-RESISTANCE

A coercion-resistant voting scheme offers not only receipt-freeness, but also defense against randomization, forced-abstention and simulation attacks. Generally, the

adversary may force the targeted voters to reveal their secret keys after registration phase, or may specify that these voters cast a special ballot. If the adversary can determine whether or not voters behaved as ordered, then the adversary is capable of coercion or otherwise actions under influence over the election process. Hence a coercion-resistant voting system is one in which the user can make the adversary into thinking that she has behaved following the order, when the voter has in fact cast a ballot according to her own willingness. So if adversary cannot distinguish between the output from a vote of her choice and any vote of the voter's choice, then the voting scheme is coercion-resistance. Here our formal definition considers receipt-freeness, forced-abstention and simulation attacks.

DEFINITION: (COERCION-RESISTANCE)

An election context ElectionContext guarantees coercion-resistance of the internet voting protocol if there exists the observational equivalence: ElectionContext [*]≈ ElectionContext [0].

Where:

$$* \equiv \text{Voter}^{\text{honest}} \mid \text{Voter}_{\text{vaild}}^{\text{attacker}} \mid \text{Voter}_{\text{vaild}}^{\text{cocered}} \mid \text{Voter}_{\text{abs}}^{\text{attacker}} \mid \text{Tally}_{\text{Auth}}$$

$$o \equiv \text{Voter}^{\text{honest}} \mid \text{Voter}_{\text{fake}}^{\text{attacker}} \mid \text{Voter}_{\text{fake}}^{\text{cocered}} \mid \text{Voter}_{\text{abs}}^{\text{attacker}} \mid \text{Tally}_{\text{Auth}}$$

Intuitively, an internet voting protocol MainP is coercion-resistance if it is observationally ElectionContext[*] equivalent to ElectionContext[o], that is meaning that the election context can not distinguish the ballots associated with valid credentials and ballot associated with fake credentials.

If ElectionContext[*]≈ElectionContext [o] is true then election context ElectionContext[] cannot distinguish between the output from a vote of her choice and any vote of the voter's choice, that is mean, the attacker can not find the difference between the fake ballots and the valid ballots, then he can not distinguish the receipt of the fake ballots and the valid ballots according the structure of receipt, in other words, the voter have the ability generating a fake receipt that can be verified by attacker, thus the internet voting protocol is receipt-freeness; If the adversary gets the credential identity of coerced voter and impersonates the coercer voter to vote his favor ballot or force the coerced voter to abandon vote, owning to observational equivalence ElectionContext [*]≈ ElectionContext [o], the tallying authority process:

$$\text{Tally}_{\text{Auth}} \equiv \text{context}\left[\begin{array}{l}\text{CcredStatus}[i]\langle\text{credStatus}:\text{Tstatus}\rangle;\\ \text{Tallying};\overline{\text{Ctallyvote}[i]}\langle\text{ballot}:\text{Tballot}\rangle\end{array}\right]$$

can use the variables credStatus to find the ballots voted by the attacker, then those ballots are eliminated, thus it does affect the soundness of result of tallying, hence it is against forced-abstention and simulation attacks .

According to definition, we can use the proposed method that prove the observational equivalence between the two processes have the same structure and differ only by the terms and term evaluations that they contains in computational model to prove the coercion-resistance in internet voting protocols.

CONCLUSION

During the past few decades internet voting protocols has been studied deeply. A lot of internet voting protocols have been developed which claimed that have the security properties. In this paper we firstly review the state-of-art of internet voting protocols and its proof in symbolic model and in computational model. Then we also talk about CryptoVerif and propose a method to prove the observational equivalence between the two processes have the same structure and differ only by the terms and term evaluations in computational model. After that we propose the first mechanized framework of coercion-resistance in internet voting protocols in computational model with active adversary. The coercion-resistance is expressed by observational equivalence. This mechanized framework can be used to automatically analyze coercion-resistance of internet voting protocols with CryptoVerif based on the method proving observational equivalence.

Owning to our formal definition only considers receipt-freeness and forced-abstention and simulation attacks, in the future work we also deal with the randomization attacks. At the same time it is interesting work to use it to analyze several typical internet voting protocols with CryptoVerif.

ACKNOWLEDGMENTS

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. We also thank Blanchet for the discussion of CryptoVerif.

REFERENCES

Acquisti, A., 2004. Receipt-free homomorphic elections and write-in voter verified ballot. CMU-ISRI-04-116, 2004, Carnegie Mellon Institute for Software Research International. http://www.heinz.cmu.edu/~acquisti/papers/acquisti-electronic_voting.pdf

Araujo, R., S. Foulle and J. Traore, 2008. A practical and secure coercion-resistant scheme for remote elections. http://drops.dagstuhl.de/opus/volltexte/2008/1295

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

Benaloh, J. and D. Tuinstra, 1994. Receipt-free secret-ballot elections. Proceeding of the 26th Annual ACM Symposium on Theory of Computing, May 23-25, ACM, New York, USA., pp: 544-553.

Blanchet, B., 2008. A computationally sound mechanized prover for security protocols. IEEE Trans. Dependable Secure Comput., 5: 193-207.

Darmawan, N., A.Y.L. Chong, K.B. Ooi and V.A.L. Venggadasallam, 2009. Security mechanism in computer network environment: A study of adoption status in Malaysian company. J. Applied Sci., 9: 2735-2743.

Delaune, S., S. Kremer and M. Ryan, 2006a. Verifying properties of electronic voting protocols. http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/DKR-wote06.pdf

Delaune, S., S. Kremer and M.D. Ryan, 2006b. Coercion-resistance and receipt-freeness in electronic voting protocol. Proceedings of 19th IEEE Computer Security Foundations Workshop, July 5-7, Venice, Italy, pp: 28-42.

Derakhshandeh, Z., B.T. Ladani and N. Nematbakhsh, 2008. Modeling and combining access control policies using Constrained Policy Graph (CPG). J. Applied Sci., 8: 3561-3571.

Fujioka, A., T. Okamoto and K. Ohta, 1992. A practical secret voting scheme for large scale elections. Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques: Advances in Cryptology, December 13-16, 1992, Queensland, Australia, pp: 244-251.

Hashemi, M., N. Ithnin and R. Pakdel, 2012. Multi touch graphical password: Usability features. Asian J. Appl. Sci., 5 : 20-32.

Jonker, H.L. and E.P. De-Vink, 2006. Formalising receipt-freeness. Proceedings of the 9th International Conference on Information Security, Aug. 30-Sept. 2, Samos Island, Greece, pp: 476-488.

Juels, A. and M. Jakobsson, 2002. Coercion-resistant electronic elections, 2002. http://www.vote-auction.net/VOTEAUCTION/165.pdf

Juels, A., D. Catalano and M. Jakobsson, 2005. Coercion-resistant electronic elections. Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, Nov. 07-07, Alexandria, VA, USA., pp: 61-70.

Lee, B., C. Boyd, E. Dawson, K. Kim, J. Yang and S. Yoo, 2003. Providing receipt-freeness in mixnet-based voting protocols. http://caislab.icu.ac.kr/Paper/paper_files/2003/ICISC03/mnvoting-final-icisc20.pdf

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

Meng, B., 2007b. Analysis of internet voting protocols with jonker-vink receipt freeness formal model. Proceedings of the International Conference on Convergence Information Technology, Nov. 21-23, ICCIT., IEEE Computer Society, Washington, DC., pp: 663-669.

Meng, B., 2008. Formal analysis of key properties in the internet voting protocol using applied pi calculus. Inform. Technol. J., 7: 1133-1140.

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

Meng, B., 2009b. A formal logic framework for receipt-freeness in internet voting protocol. J. Comput., 4: 184-192.

Meng, B., 2009c. 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., 2009d. 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. and J.Q. Wang, 2010. An efficient receiver deniable encryption scheme and its applications. J. Networks, 5: 683-690.

Meng, B., Z. Li and J. Qin, 2010a. 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., W. Huang and J. Qin, 2010b. 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, 2010c. 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., 2011. Refinement of mechanized proof of security properties of remote internet voting protocol in applied PI calculus with proverif. Inform. Technol. J., 10: 293-334.

Mousa, A., 2005. Sensitivity of changing the RSA parameters on the complexity and performance of the algorithm. J. Applied Sci., 5: 60-63.

Nabi, M.S.A., M.L.M. Kiah, B.B. Zaidan, A.A. Zaidan and G.M. Alam, 2010. Suitability of using SOAP protocol to secure electronic medical record database transmission. Int. J. Pharmacol., 6: 959-964.

Rabah, K., 2005. Theory and implementation of elliptic curve cryptography. J. Applied Sci., 5: 604-633.

Samimi, A. and M.A. Golkar, 2012. A novel method for optimal placement of FACTS based on sensitivity analysis for enhancing power system static security. Asian J. Appl. Sci., 5 : 1-19.

Sharma, S., R.C. Jain and S.S. Bhadauria, 2007. SBEVA: A secured bandwidth efficient variance adaptive routing protocol for mobile Ad hoc network. Asian J. Inform. Manage., 1: 1-10.

Zaidan, A.A., B.B. Zaidan, A.K. Al-Frajat and H.A. Jalab, 2010. An overview: Theoretical and mathematical perspectives for advance encryption standard/rijndael. J. Applied Sci., 10: 2161-2167.