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 10 (3): 611-625, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.611.625

Computationally Sound Mechanized Proofs for Deniable Authentication Protocols with a Probabilistic Polynomial Calculus in Computational Model

Bo Meng and Fei Shao
School of Computer, South-Center University for Nationalities, MinYuan Road # 708,
Hong Shan Section, Wuhan, Hubei, 430074, China

Abstract: During the past few decades deniable authentication protocol has been studied. A lot of deniable authentication protocols have been proposed which claimed that have the security properties, for example, authentication, deniability and so on. To our knowledge, these security properties and deniable authentication protocols are analyzed with informal method or with symbolic method by hand which depends on experts' knowledge and skill and is prone to make mistakes. So analysis of security properties and deniable authentication protocols with automatic tool in symbolic model or computational model plays an important role in security protocol world and is a significant work. Especially analysis with automatic tool in computational model is a changeling issue. In this study firstly the state-of-art of deniable authentication protocol and the proof including in symbolic model and in computational model are presented. Then the term, process and correspondence assertion in Blanchet calculus is used to model security properties including deniability and deniable authentication protocols and propose the first mechanized framework of deniable authentication protocols in computational model with active adversary. The proposed mechanized framework can be used to automatically analyze the security properties including strong deniability and weak deniability of interactive or non-interactive deniable authentication protocols with CryptoVerif.

Key words: Computational model, strong deniability, weak deniability, deniable authentication protocol, automatic framework

INTRODUCTION

With the development of internet technology, many transactions are carried out through the internet. Most of transactions can not be finished just by face-to-face approach. A key problem of how to authenticate the identities of the involved parties emerges subsequently. Therefore, many authentication protocols have been presented during the past few decades. Authentication protocol based on cryptographic technologies is used to confirm the identities of parties in the communication (Meng, 2009a).

However, some specified internet applications such as electronic voting and electronic business, require deniable authentication protocols. Deniable authentication protocol allows a sender to authenticate a message for a receiver, in a way that the receiver can not convince a third party that such authentication (or any authentication) ever took place. Deniable authentication protocol has two characteristics that differ from traditional authentication protocol. One is that only the intended receiver can authenticate the true source of a given message. The other is that the sender can not provide the evidences to prove the source of the message to a third party in some condition and the receiver can provide the evidences to prove the source of the message to a third party (Raimondo and Gennaro, 2005).

The practical secure deniable authentication protocol should have the following properties: completeness or authentication; strong deniability (Raimondo and Gennaro, 2005); weak deniability (Raimondo and Gennaro, 2005); security of forgery attack (Shao, 2004); security of impersonate attack (Shao, 2004); security of compromising session secret attack (Lee *et al*., 2007); security of man-in-the-middle attack (Han *et al*., 2005).These secure properties play an important role in implementation of secure transactions over the public internet.

During the past few decades deniable authentication protocol has been studied. Deniable authentication protocol falls into two categories: Interactive deniable authentication protocol (Aumann and Rabin, 1998; Dwork *et al*., 1998; Deng *et al*., 2001; Fan *et al*., 2002; Raimondo and Gennaro, 2005; Han *et al*., 2005; Feng and Ma, 2007) and non-interactive deniable authentication

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

protocol (Shao, 2004; Lu and Cao, 2005a, b; Qian *et al*., 2005; Shi and Li, 2005; Lee *et al*., 2007; Meng, 2009a).

In order to verify the security properties of security protocol including deniable authentication protocols and increase the confidence of the people, two approaches have been developed for analyzing security protocols form the beginning of the 1980s. One approach relies on a symbolic model of protocol executions in which cryptographic primitives are treated as black boxes. Since the seminal work of Dolev and Yao, it has been realized that this approach enables significantly simpler and many automatic tools have been developed. For example, SMV, NRL, Casper, Isabelle, Athena, Revere, SPIN, Brutus, ProVerif (Blanchet, 2001), Scyther. In symbolic model, recently, Meng (2009b) proposes a framework of strong and weak deniability based on Kessler and Neumann logic is proposed. After that, the framework is applied to analyze the deniability of two typical deniable authentication protocols: Fan *et al*. (2002) interactive deniable authentication protocol and Meng non-interactive deniable authentication protocol. But the result of proof based on symbolic model is not quite clear.

The other approach relies on a computational model that base issues of complexity and probability. This approach use a strong notion of security, guaranteed against all probabilistic polynomial-time attacks. The computation approach can be more realistic, but it is difficult to automatic proof until the introduction of mechanized tool CryptoVerif (Blanchet, 2005, 2006, 2007a, b; Blanchet *et al*., 2008) which is the only automatic tool with computational model.

To our best knowledge, these security properties and deniable authentication protocols are analyzed with informal method, or with symbolic method by hand (Meng, 2009b), which depends on experts' knowledge and skill and is prone to make mistakes. So analysis of security properties and deniable authentication protocols with automatic tool in symbolic model or computational model plays an important role in security protocol world and is a significant work. Especially analysis of security properties and deniable authentication protocols with automatic tool in computational model is a changeling issue.

Recently, Blanchet (2005, 2006, 2007a, b) propose a probabilistic polynomial calculus based on the pi calculus and the calculi introduced in Lincoln *et al*. (1998, 1999), Laud (2005) and Micciancio and Warinschi (2004) based on computational model. In this calculus, messages are bitstrings and cryptographic primitives are functions operating on bitstrings. Blanchet calculus is adapted from the pi calculus and its semantics is purely probabilistic (no non-determinism). All processes run in polynomial time: polynomial number of copies of processes and length of messages on channels bounded by polynomials. Blanchet calculus has been carefully designed to make the automated proof security protocols. At the same time they develop a mechanized tool CryptoVerif (Blanchet, 2005, 2006, 2007a, b) which is the only automatic tool with computational model until now. CryptoVerif does not rely on soundness results for symbolic model but directly automate the proofs made in cryptography, based on sequences of games. It 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 can prove secrecy properties and events that can be executed only with negligible probability; also it can handle 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. It can also give a bound on the probability of an attack. CryptoVerif runs either automatically or interactively, in which case it receives guidance from the user for selecting transformations. In a recent case study, CryptoVerif is used to verify: FDH signature scheme (Blanchet and Pointcheval, 2006) PKINIT for Kerberos (Jaggard *et al*., 2007), Verification Protocol Implementations in ML (Bhargavan *et al*., 2007), a model of the Basic and Public-Key Kerberos protocol (Blanchet *et al*., 2008), Verification Protocol Implementations for TLS (Bhargavan *et al*., 2008), Diffie-hellman protocol.

Inspired by the work of Blanchetand owning to the introduction of the powerful mechanized tool CryptoVerif which is the only automatic tool in computational model and has been successfully used to verify several cryptographic primitive and security protocols, in this study, we use Blanchet calculus in the computational model to analyze the typical deniable authentication protocols with mechanized tool CryptoVerif. The main contributions of this paper are summarized as follows:

- The state-of-art of deniable authentication protocol and the proof including in symbolic model and in computational model are presented. We find that security properties and deniable authentication protocols are analyzed with informal method or with symbolic method by hand
- The first mechanized framework of deniable authentication protocol is proposed based on Blanchet calculus in computational model with active adversary. This mechanized framework can be used to automatically analyze the security properties including strong deniability and weak deniability of interactive deniable authentication protocols and non-interactive deniable authentication protocols

CONTRIBUTION AND OVERVIEW

During the past few decades deniable authentication protocol has been studied. A lot of deniable authentication protocols have been proposed which claimed that have the security properties, for example, authentication; strong deniability (Raimondo and Gennaro, 2005); weak deniability (Raimondo and Gennaro, 2005); security of forgery attack (Shao, 2004); security of impersonate attack (Shao, 2004); security of compromising session secret attack (Lee *et al*., 2007); security of man-in-the-middle attack (Han *et al*., 2005) and so on. In order to verify the security properties of security protocol including deniable authentication protocols and increase the confidence of the people, two approaches have been developed for analyzing security protocols form the beginning of the 1980s. One approach relies on a symbolic model of protocol executions in which cryptographic primitives are treated as black boxes. The other approach relies on a computational model that base issues of complexity and probability. This approach use a strong notion of security, guaranteed against all probabilistic polynomial-time attacks. The computation approach can be more realistic. To our best knowledge, these security properties and deniable authentication protocols are analyzed with informal method, or with symbolic method by hand (Meng, 2009b), which depends on experts' knowledge and skill and is prone to make mistakes.

So analysis of security properties and deniable authentication protocols with automatic tool in symbolic model or computational model plays an important role in security protocol world and is a significant work. Especially analysis of security properties and deniable authentication protocols with automatic tool in computational model is a changeling issue.

Recently owning to the introduction of a probabilistic polynomial calculus proposed by Blanchet (2005, 2006, 2007a, b) and a powerful mechanized tool CryptoVerif (Blanchet, 2005, 2006, 2007a, b) which is the only automatic tool in computational model are have been successfully used to verify several cryptographic primitive and security protocols, we use Blanchet calculus in the computational model to analyze the typical deniable authentication protocols with CryptoVerif. The main contributions of this paper are summarized as follows in detail:

Firstly the state-of-art of deniable authentication protocol and the proof including in symbolic model and in computational model are presented. We find that security properties and deniable authentication protocols are analyzed with informal method or with symbolic method by hand.

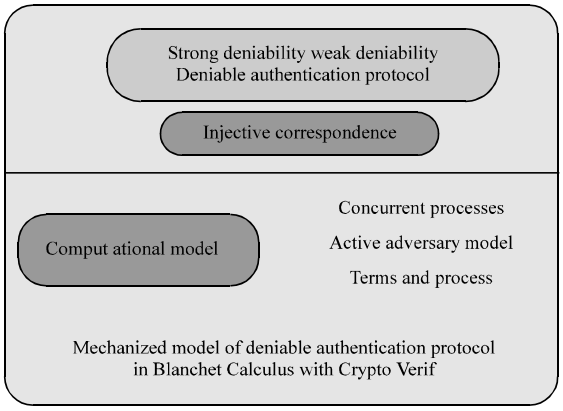

Fig. 1: Analysis model of deniable authentication protocols with Blanchet calculus

Then the term, process and correspondence assertion in Blanchet calculus is used to model the security properties including strong deniability and weak deniability and deniable authentication protocol. Finally we propose the first mechanized framework of deniable authentication protocols in computational model with active adversary. The strong deniability and weak deniability are expressed by non-injective or injective correspondence. This mechanized framework can be used to automatically analyze the security properties including strong deniability and weak deniability of interactive deniable authentication protocols and non-interactive deniable authentication protocols with CryptoVerif. Figure 1 describes the analysis model of deniable authentication protocols with Blanchet calculus.

RELATED WORK

Deniable authentication protocols allow a sender to authenticate a message for a receiver, in a way that the receiver can not convince a third party that such authentication (or any authentication) ever took place. Deniable authentication has two characteristics that differ from traditional authentication: one is that only the intended receiver can authenticate the true source of a given message; the other is that the receiver can not provide the evidences to prove the source of the message to a third party.

A practical secure deniable authentication protocol should have the following properties: Completeness or authentication; strong deniability (Raimondo and Gennaro, 2005); weak deniability (Raimondo and Gennaro, 2005); security of forgery attack (Shao, 2004); security of impersonate attack (Shao, 2004); security of compromising session secret attack (Lee *et al*., 2007); security of man-

in-the-middle attack (Han *et al*., 2005).These secure properties play an important role in implementation of secure transactions over the public Internet.

During the past few decades deniable authentication protocol has been studied. Deniable authentication protocol falls into two categories: Interactive deniable authentication protocol (Aumann and Rabin, 1998; Dwork *et al*., 1998; Deng *et al*., 2001; Fan *et al*., 2002; Raimondo and Gennaro, 2005; Han *et al*., 2005; Feng and Ma, 2007) and non-interactive deniable authentication protocol (Shao, 2004; Lu and Cao, 2005a, b; Qian *et al*., 2005; Shi and Li, 2005; Lee *et al*., 2007; Meng, 2009a).

Dwork *et al*. (1998) proposes an interactive deniable authentication protocol based on the concurrent zero-knowledge proof. Aumann and Rabin (1998) propose an interactive deniable authentication protocol based on factoring problem. Deng *et al*. (2001) propose two interactive deniable authentication protocols based on factoring and the discrete logarithm problem, respectively. Fan *et al*. (2002) propose another simple interactive deniable authentication protocol based on the Diffie-Hellman key distribution protocol. Han *et al*. (2005) propose an interactive deniable authentication protocol resisting man-in-the-middle attack based on Diffie-Hellman key exchange protocol. Feng and Ma (2007) propose a concurrent deniable authentication based on witness indistinguishable which can support strong deniability.

The interactive deniable authentication protocols are inefficient. Hence several non-interactive deniable authentication protocols are proposed. Shao (2004) points out there are three weakness in paper (Fan *et al*., 2002) and give an improved a generalized ElGamal signature scheme. Lu and Cao (2005a, b) propose a non-interactive deniable authentication protocol based on bilinear pairings and factoring, respectively. Lee *et al*. (2007) point that protocols (Shao, 2004; Lu and Cao, 2005a, b) can not protect against compromising session secret attack and introduce a new deniable authentication protocol using generalized ElGamal signature scheme. But these non-interactive deniable authentication protocols have not strong deniability. Meng (2009a) proposes a secure non-interactive deniable authentication protocol based on discrete logarithm problem is developed. At the same time we prove that the proposed protocol has properties: completeness, strong deniability, weak deniability, security of forgery attack, security of impersonate attack, security of compromising session secret attackand security of man-in-the-middle attack. These deniable authentication protocols are analyzed with informal method.

In order to verify the security properties of security protocol including deniable authentication protocols and increase the confidence of the people, two approaches have been developed for analyzing security protocols form the beginning of the 1980s. The one approach relies on a symbolic model of protocol executions in which cryptographic primitives are treated as black boxes. Since the seminal work of Dolev and Yao, it has been realized that this approach enables significantly simpler and many automatic tools have been developed. But the result of proof is not quite clear. The other approach relies on a computational model that base issues of complexity and probability. This approach use a strong notion of security, guaranteed against all probabilistic polynomial-time attacks. The computation approach is more realistic, but it is difficult to implement with automatic proof.

In symbolic model Meng (2009b) proposes a framework of strong and weak deniability based on Kessler and Neumann logic is proposed. After that, the framework is applied to analyze the deniability of two typical deniable authentication protocols: Fan *et al*. (2002) interactive deniable authentication protocol and Meng non-interactive deniable authentication protocol by hand.

To our best knowledge, deniable authentication protocols have not been analyzed in computational model.

Recently Blanchet (2005, 2006, 2007a, b) proposes a probabilistic polynomial calculus which is inspired by the pi calculus and the calculi introduced in Lincoln *et al*. (1998, 1999), Laud (2005) and Micciancio and Warinschi (2004) based on computational model. In this calculus, messages are bitstrings and cryptographic primitives are functions operating on bitstrings. Blanchet calculus is adapted from the pi calculus and its semantics is purely probabilistic (no non-determinism).All processes run in polynomial time: polynomial number of copies of processes and length of messages on channels bounded by polynomials. Blanchet calculus has been carefully designed to make the automated proof security protocols. Blanchet calculus consists of terms and processes. At the same time they develop a mechanized tool CryptoVerif (Blanchet, 2005, 2006, 2007a, b) with computational model. CryptoVerif does not rely on soundness results for symbolic model but directly automate the proofs made in cryptography, based on sequences of games. It 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 can prove secrecy properties and that events can be executed only with negligible probability, also it can handle 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. It can also give a bound on the probability of an attack (exact security). CryptoVerif runs

either automatically or interactively, in which case it receives guidance from the user for selecting transformations. In a recent case study, CryptoVerif is used to verify: FDH signature scheme (Blanchet and Pointcheval, 2006) PKINIT for Kerberos (Jaggard *et al*., 2007), Verification Protocol Implementations in ML (Bhargavan *et al*., 2007), a model of the Basic and Public-Key Kerberos protocol (Blanchet *et al*., 2008), Verification Protocol Implementations for TLS (Bhargavan *et al*., 2008), Diffie-hellman protocol.

Inspired by the works of Blanchet, to our best knowledge, deniable authentication protocols have not been analyzed in computational model, in this study, we use Blanchet (2005, 2006, 2007a, b) in the computational model to analyze the typical deniable authentication protocols with CryptoVerif (Blanchet, 2005, 2006, 2007a, b; Blanchet *et al*., 2008).

REVIEW OF BLANCHET CALCULUS

Here, we review Blanchet calculus (Blanchet, 2005, 2006, 2007a, b) based on the pi calculus and the calculi introduced in Lincoln *et al*. (1998, 1999), Laud (2005) and Micciancio and Warinschi (2004) based on computational model. Blanchet calculus is a probabilistic polynomial calculus and has been carefully designed to make the automated proof security protocols. In this calculus, messages are bitstrings and cryptographic primitives are functions operating on bitstrings. Blanchet calculus is adapted from the pi calculus and its semantics is purely probabilistic (no non-determinism).All processes run in polynomial time: Polynomial number of copies of processes and length of messages on channels bounded by polynomials. Blanchet calculus consists of terms and processes. Before introduction of its syntax and informal semantics we firstly discus the notations used in Blanchet calculus.

Notations: The substitution that replaces x_j with M_j for each $j \leq m$ is denoted by:

$$\left\{ M_1 / x_1, \ldots, M_m / x_m \right\}$$

the cardinal of a set or multiset s is denoted by $|s|$. If s is a finite set,

$$x \xleftarrow{R} S$$

chooses a random element uniformly in s and assigns it to x. If A is a probabilistic algorithm, the experiment of choosing random coins r and assigning to x the result of running $A(x_1, \ldots, x_m)$ with coins r is denoted by $x \leftarrow A(x_1, \ldots, x_m)$. otherwise, $x \leftarrow M$ is a simple assignment statement.

Syntax and informal semantics: In Blanchet calculus denote a countable set of channel names. The mapping from channels to integers is denoted by maxlen η. maxlen $\eta(c)$ is the maximum length of a message sent on channel c. Longer messages are truncated. For all c, maxlen $\eta(c)$ is polynomial in η. $I\eta(n)$ means that the interpretation of n for a given value of the security parameter η and is a polynomial bounded, efficiently computable function of η. For each value of the security parameter η, each type corresponds to a subset $I\eta(T)$, where T is the type, of Bitstring $\cup \{\perp\}$ where Bitstring is the set of all bitstrings and $\perp$ is a special symbol. The set $I\eta(T)$ must be recognizable in polynomial time, that is, there exists an algorithm that decides whether $x \in I\eta(T)$ in time polynomial in the length x of and the value of η. Each function symbol f comes with a type declaration $f: T1 \times \ldots \times Tm \rightarrow T$. For each value of η, each function symbol f corresponds to a function $I\eta(f)$ from $I\eta(T1) \times \ldots \times I\eta(T)$ to, such that $I\eta(f)(x_1, \ldots, x_m)$ is computable in polynomial time in the lengths of $x_1, \ldots, x_m$ and the value of η. Terms in Fig. 2 in Blanchet calculus represent computations on bitstrings. The replication index is an integer which serves in distinguishing different copies of a replicated process $!^{i \leq n}$. The variable access $x[M_1, \ldots, M_m]$ returns the content of the cell indices $M_1, \ldots, M_m$ of the m-dimensional array variable x. Generally x, y, z denote as variable names. The function application $f(M_1, \ldots, M_m)$ returns the result of applying function f to $M_1, \ldots, M_m$. Using different channels for each input and output allows the adversary applying function f to $M_1, \ldots, M_m$. Using different for each input and output allows the adversary to control

M, N ::=	terms
i	replication index
$x[M_1, \ldots, M_m]$	variable access
$f[M_1, \ldots, M_m]$	function application

Fig. 2: Terms

Q ::=	input process
0	nil
Q|Q′	paralell composition
$!^{i \leq n}Q$	replication times
new channel c; Q	channel restriction
$c[M_1, L, M_l](x_1[\tilde{i}] : T_1, L, x_k[\tilde{i}] : T_k); P$	input

Fig. 3: Input process

$P ::=$	Output process
$\overline{c\left[M_1,\cdots M_l\right]}\left\langle N_1,\cdots,N_k\right\rangle;Q$	Out put
$\text{new } x\left[i_1,\cdots,i_m\right]: T;P$	Random number
$\text{let } x\left[i_1,\cdots,i_m\right]: T = M \text{ in } P$	Assignment
$\text{if defined}\left(M_1,\cdots M_l\right)\wedge M \text{ then P else } P'$	Conditional
$\text{find } \left(\oplus_{j=1}^{m} u_{j1},\cdots,u_{jm_j}\left[\tilde{i}\right]\le n_{jm_j} \text{suchthat defined}\left(M_{j1},\cdots,M_{jlj}\right)\wedge M_j \text{ then } P_j\right) \text{ else P}$	Array lookup
$\text{event e}\left(M_1,\cdots,M_m\right);P$	Event

Fig. 4: Output process

the network. Variables can be defined by assignments, inputs, restrictions and array lookups. The find construct allows us to access arrays, which is the key for its purpose. Using arrays allows us to remember the values of the variables in each copy of the processes, so that the whole state of the system is available. In Blanchet calculus, arrays replace lists often used by cryptographers in their proofs.

In Blanchet calculus process consists of two kinds of processes: input processes Q in Fig. 3 are ready to receive a message on a channel; output processes P in Fig. 4 output a message on a channel after executing some internal computations. The input process 0 does nothing; $Q|Q'$ is the parallel composition of Q and Q′; $!^{i\le n}Q$ represents copies n of Q in parallel, each with a different value of i∈[1, n] new channel c; Q creates a new private channel n and executes. The output process new x i_1, ..., i_m: T chooses a new random number uniformly in Iη (T), stores it in i_1, ..., i_m and executes P. Function symbols represent deterministic functions, so all random numbers must be chosen by new x 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 x i_1, ..., i_m: T = M in P stores the bitstring value of M in new x i_1, ..., i_m and executes P:

$$\text{find } \left(\begin{array}{l}\oplus_{j=1}^{m} u_{j1},\cdots,u_{jm_j}\left[\tilde{i}\right]\le n_{jm_j} \text{ suchthat defined}\\ \left(M_{j1},\cdots,M_{jlj}\right)\wedge M_j \text{ then } P_j\end{array}\right) \text{ else P}$$

means that it tries to find a branch J in [1, m] such that there are values of uj1, ..., ujmj for which Mj1, ..., Mjlj are defined and Mj 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 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.

The adversary is presented by an evaluation context c. An evaluation context is a process with a $hole_{xl,\ldots,xm[]}$, of one of the following forms: A hole, a process in parallel with an evaluation context Q|C, or a restriction new channel c; C, which limits the scope of the channel c to the context C. C [Q] present the process obtained by replacing the hole of c with Q. When v is a set of variables defined in Q, an evaluation context c is said to be acceptable for (Q, V) if and only if c does not contain events, the common variables of c and Q are in V and C [Q] satisfies the well-formed invariants. The set V contains the variables the context is allowed to access.

Formal semantics: The semantics is defined by a probabilistic reduction relation. For each process Q, there exists a probabilistic polynomial time Turing machine that simulates Q. Processes run in polynomial time since the number of processes created by a replication and the length of messages sent on channels are bounded by polynomials. Conversely, Blanchet calculus can simulate a probabilistic polynomial-time Turing machine, simply by choosing coins by new and by applying a function symbol defined to perform the same computations as the Turing machine.

Observational equivalence: Two processes which are also games Q^1, Q^2 are observationally equivalent when the adversary has a negligible probability of distinguishing them. The adversary is represented by an acceptable evaluation context c. A context is a process containing a $hole_{x_l}$, ..., $x_{m[]}$. An evaluation context c is a context built from x_l, ..., $x_{m[]}$, new channel c; C, Q|C and C|Q, an evaluation context represent the adversary. The process obtained by replacing the $hole_{[]}$ in the context c with the process Q is denoted by C [Q]. Essentially, a process put in parallel with the considered games.

Definition: Let Q^1 and Q^2 be two processes and v a set of variables. Assume that Q^1 and Q^2 satisfy Invariants 1, 2 and 3 and the variables of v are defined in Q^1 and Q^2, with the same types. An evaluation context is said to be acceptable for Q^1 and, v if and only if v:

$$\mathrm{var}(C) \cap \left(\mathrm{var}(Q^1) \cup \mathrm{var}(Q^2)\right) \subseteq V$$

and C [Q^1] satisfies Invariants 1, 2and 3. We say that Q^1 and Q^2 are observationally equivalent with public variables v, written $Q^1 \approx^v Q^2$, when for all evaluation contexts acceptable for Q^1, Q^2, v, for all channels c and bitstrings a:

$$\left|\Pr\left[C(Q^1) \rightsquigarrow_\eta \overline{c}\langle a\rangle\right] - \Pr\left[C(Q^2) \rightsquigarrow_\eta \overline{c}\langle a\rangle\right]\right|$$

is negligible.

Intuitively, the goal of the adversary represented by context c is to distinguish Q^1 and Q^2. When it succeeds, it performs a different output. When $Q^1 \approx^v Q^2$, the context has negligible probability of distinguishing Q^1 and Q^2.

CORRESPONDENCES ASERTIONS

In security protocol, correspondence assertions can be used to describe the properties of the form if some events have been executed, then some other events have been executed, where each event corresponds to a certain point in the protocol, possibly with arguments. Correspondences consist of non-injective and injective correspondences. A non-injective correspondence is a property of the form if some events have been executed, then some other events have been executed at least once. Injective correspondences are properties of the form if some event has been executed n times, then some other events have been executed at least n times. The definitions of non-injective and injective correspondences (Blanchet, 2007a, b) based on Blanchet calculus are introduced.

Non-injective correspondences: A non-injective correspondence is a property of the form if some events have been executed, then some other events have been executed at least once. Here, Blanchet generalize these correspondences to implications between logical formula $\psi \Rightarrow \phi$, which may contain events.

Definition 1: The sequence of events ε satisfies the correspondence $\psi \Rightarrow \phi$, written:

$$\varepsilon \mapsto \psi \Rightarrow \phi$$

if and only if for all ρ defined on var (ψ) such that $\rho', \epsilon \mid\!- \psi$ there exist an extension ρ' of ρ to var (ϕ) such that $\rho', \epsilon \mid\!- \psi$.

Intuitively, a sequence of events ϵ satisfies $\psi \Rightarrow \phi$ when, if ϵ satisfies ψ and then ϵ satisfies ϕ. The variables of ψ are universally quantified; those of ϕ that do not occur in ψ are existentially quantified.

Definition 2: The process Q satisfies the correspondence $\psi \Rightarrow \phi$ with public variables v if and only if for all evaluation contexts c acceptable for (Q, V):

$$\psi \Rightarrow \phi \Pr\left[\exists(\mathbb{C}, \varepsilon), \mathrm{initConfig}(C[Q]) \xrightarrow{\varepsilon} \mathbb{C} \wedge \varepsilon \vee \psi \Rightarrow \phi\right]$$

is negligible.

Intuitively, a process satisfies $\psi \Rightarrow \phi$ when the probability that it generates a sequence of events that does not satisfy $\psi \Rightarrow \phi$ is negligible, in the presence of an adversary represented by the context C.

Injective correspondences: Injective correspondences are properties of the form if some event has been executed n times, then some other events have been executed at least n times. The grammar of formulae ϕ with injective events inj-event) (e (M_1, ..., M_2)) is also included in Blanchet calculus. The formula ϕ is a conjunction of events. The conditions on the number of executions of events apply only to injective events.

Definition: The sequence of events ϵ satisfies the correspondence $\psi \Rightarrow \phi$:

$$\varepsilon \mapsto \psi \Rightarrow \phi$$

written, if and only if there exists a component-wise injective F such that for all ρ defined on var (ψ), for allψ^T such that:

$$\rho, \varepsilon \mid\!- \psi^T \psi$$

there exists an extension ρ' of ρ to var (ϕ) such that:

$$\rho', \varepsilon \mid\!- F(\psi^T) \Rightarrow \phi$$

Intuitively, a sequence of events ϵ satisfies $\psi \Rightarrow \phi$ when, if ϵ satisfies ψ with execution steps defined by ψ^T, then ϵ satisfies ϕ with execution steps defined by F (ψ^T).

The injectivity is guaranteed because F is component-wise injective.

MECHANIZED PROOF TOOL CRYPTOVERIF

Here, we give a brief overview of the mechanized prover CryptoVerif, formalize deniable authentication protocol using itand summarize authentication and secrecy properties proved by CryptoVerif (http://www.cryptoverif.ens.fr). In most cases, the prover succeeds in proving the desired properties when they holdand it always fails to prove them when they do not hold. In other words CryptoVerif is sound and is not complete. Hence it cannot prove some security properties which are valid.

The mechanized prover CryptoVerif (Blanchet, 2005, 2006, 2007a, b) 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 prove secrecy properties and events that can be executed only with negligible probability; also it can handle 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. It can also give a bound on the probability of an attack (exact security). CryptoVerif runs either automatically or interactively, in which case it receives guidance from the user for selecting transformations. In a recent case study, CryptoVerif is used to verify: FDH signature scheme(Blanchet and Pointcheval,2006) PKINIT for Kerberos (Jaggard *et al.* 2007), Verification Protocol Implementations in ML (Bhargavan *et al.*, 2007), a model of the basic and public-key Kerberos protocol (Blanchet *et al.*, 2008), Verification Protocol Implementations for TLS (Bhargavan *et al.*, 2008), Diffie-hellman protocol.

Secrecy properties which consist of one-session secrecy and secrecy can be expressed as indistinguishability between two configurations. CryptoVerif has the ability to analyze the two notions of secrecy. The one-session secrecy (query secret1 in CryptoVerif) states that the adversary cannot distinguish the value of any instance of a given variable from a random value, the adversary perform a single query; the secrecy (query secret in CryptoVerif) states that the adversary cannot distinguish the vector of values of all the instances of a given variable from a vector of independent random values, in this definition the adversary can perform several test queries.

Authentication is expressed as correspondences (Blanchet, 2007a, b), much as in symbolic models. Computationally, correspondences assert that, if some event is executed, then other events must also have been executed, at least once, with matching parameters, at least with overwhelming probability. CryptoVerif can deal with more general properties expressed as logical formulas; also both injective and non-injective properties can be analyzed. A non-injective correspondence is a property of the form "if some events have been executed, then some other events have been executed at least once". Injective correspondences are properties of the form "if some event has been executed n times, then some other events have been executed at least n times". Injective correspondences are more difficult to check than non-injective ones, because they require distinguishing between several executions of the same event.

The mechanized prover CryptoVerif provides a generic mechanism for specifying the security assumptions on cryptographic primitives, which can handle in particular symmetric encryption, message authentication codes, public-key encryption, signatures, hash functions. The generated proofs are proofs by sequences of games, as used by cryptographers. These proofs are valid for a number of session's polynomial in the security parameter, in the presence of an active adversary. CryptoVerif can also evaluate the probability of success of an attack against the protocol as a function of the probability of breaking each cryptographic primitive and of the number of sessions (exact security).

FORMALIZING DENIABILITY IN COMPUTATIONAL MODEL

Deniable authentication protocol and active adversary model: Deniable authentication protocol in Fig. 5 is a message-driven protocol which is an iterative process described as follows. Message-driven protocol (Bellare *et al.*, 1998; Canetti and Krawczyk, 2001, 2002) consists of several parties/principles, for example, sender, receiverand so on. At the same time adversary is need to considered which are probabilistic polynomial time machines. These principles are communicated by channel by which message can be exchanged. Protocols run concurrently by these principles. The message-driven protocol is executed by a party, may be sender, with some initial state including the relative input of the protocol, for example, the protocol's inputand the receiver's identity. Once invoked by the sender or other party, the protocol generally waits for the two events: one is that the arrival of a message from the channel, the other is that an external request. Upon receiving the one of the events, the protocol processes the incoming message together with its current internal state, generates a new internal state and outgoing messages to the channel and external requests to other protocols run by the other party, then it send the message to the other party, for example, receiver. After that it waits for the next message or event to arrive.

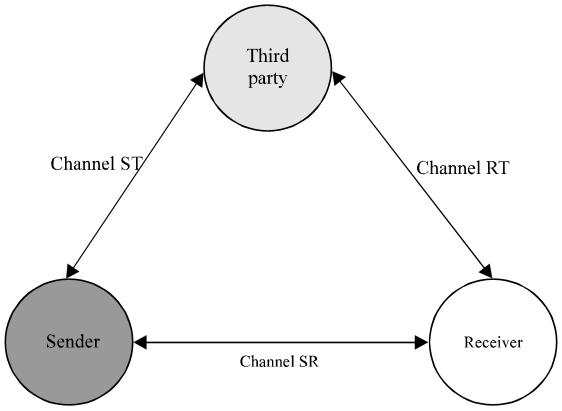

Fig. 5: The idea of deniable authentication protocol

In deniable authentication protocol the principles consist of sender, receiver and third party. Deniable authentication protocol allows a sender to authenticate a message for a receiver, in a way that the receiver can not convince a third party that such authentication (or any authentication) ever took place. Deniable authentication protocol has two characteristics that differ from traditional authentication protocol. One is that only the intended receiver can authenticate the true source of a given message. The other is that the sender can not provide the evidences to prove the source of the message to a third party in some condition and the receiver can provide the evidences to prove the source of the message to a third party. The communication channels consist of three channels. One is the channel channel SR between the sender and the receiver. The other two channels are the channel channel RT between the receiver and the third party and the channel channel ST between the sender and the third party. The channel channel ST is only used in proof of strong deniability. The channel channel RT is only used in proof of weak deniability

In order to analyze deniable authentication protocol with computational model, we consider a probabilistic polynomial-time attacker that has full control of the communications channel channel SR: It can listen to all the transmitted information, decide what messages will reach their destination and when, change these messages at will or inject its own generated messages. The formalism represents this ability of the attacker by letting the adversary be the one in charge of passing messages from one party to another. The attacker also controls the scheduling of all protocol events including the initiation of protocols and message delivery. According to the requirement of deniable authentication protocol the adversary can not control the channels channel RT and channel ST because the receiver just sends the genius message and fake message to the third party or the sender just send the relative information to the third party. The third party just checks the message received from the receiver. Deniable authentication protocols are in a context in which the honest participants are willing to run sessions with the adversary. That is mean the adversary is an active attacker in the channel channel SR and is a passive attacker in channel channel RT and channel ST.

Formalizing the deniable authentication protocols: Here, we give the formal definition of deniable authentication processes based on idea (Blanchet, 2007a) with Blanchet calculus. At the same time we also give additional explanations on our definition. Generally deniable authentication protocol includes three roles, sender which is initiator, receiver which is responder and third party, represented by Sender, Receiver and Third party, respectively. We assume Sender that plays only a role of the initiator, Receiver plays only the role of responder, Third party plays only on the prover. Deniable authentication protocol consists of a sequence of messages exchanged between the Receiver and the Receiver and Third party and Sender and Third party. In deniable authentication protocol Sender can authenticate a message for Receiver, in a way that the Receiver can not convince a Third party that such authentication (or any authentication) ever took place. Deniable authentication protocol has two characteristics that differ from traditional authentication protocol. One is that only the intended Receiver can authenticate the true source of a given message. The other is that the Sender can not provide the evidences to prove the source of the message to a third party at some condition and the Receiver can provide the evidences to prove the source of the message to a third party. The ability of adversary is defined in the previous section. It can control the channel channel SR between Sender and Receiver. It can not control the channels channel ST and channel RT. At the same time the adversary is a probabilistic polynomial-time attacker.

In our model, we assume that shared-key encryption is modeled as encrypt-then-MAC, where the encryption is indistinguishability under chosen plaintext attacks and the MAC is enforceability under chosen message attacks; public-key encryption is assumed to be indistinguishability under adaptive chosen ciphertext attacks; signatures are assumed to be unforgeability under chosen message attacks. CryptoVerif has the decisional Diffie-Hellman assumption and Computational Diffie-Hellman assumption.

In the following we give the definition of deniable authentication protocol represented by a process in Blanchet calculus.

Definition DAP: A secure deniable authentication protocol with session functions sessionid and sessionid process DAP for any probabilistic polynomial-time adversary:

$$\text{DAP} = \text{Init process};(!^{i_{\text{sender}} \le n}\ \text{Sender Process} \mid !^{i_{\text{receiver}} \le n}\ \text{Receiver Process} \mid !^{i_{\text{thirdparty}} \le n}\ \text{Third party Process})$$

Such that:

- If the adversary just send Receiver to Sender process^{i} as the first message and relays faithfully between process Sender process^{i} and process Receiver process$^{i'}$, then Receiver process$^{i'}$ process finishes with Sender and process Sender process^{i} finishes with Receiver
- With overwhelming probability, there exists an injective function that maps each index of a process Sender process^{i} that finished with to the index i' of a process with intended principle Sender such that sessioner:

$$\text{sessioner}^{1}\left(x_1[i], x_2[i], \cdots, x_l[i]\right) = \text{sessioner}^{1}\left(z_1[i'], z_2[i'], \cdots, z_l[i']\right)$$

- With overwhelming probability, there exists an injective function that maps each index i of a process Receiver process^{i} that finished with Sender to the index i' of a process that finished with Sender process$'$ such that Sessioner:

$$\text{sessioner}^{1}\left(x_1[i], x_2[i], \cdots, x_l[i]\right) = \text{sessioner}^{1}\left(z_1[i'], z_2[i'], \cdots, z_l[i']\right)$$

- If the adversary just send Third party to Receiver process^{i} as the first message and relays faithfully between Third party process^{i} and Receiver process$^{i'}$, then Third party process$^{i'}$ finishes with Receiver and Receiver process$^{i'}$ finishes with Third party^{i}
- With overwhelming probability, there exists an injective function that maps each index i of a process Third party process^{i} that finishes with Receiver to the index i' of a process Receiver process^{i} finishes with Third party such that Sessioner:

$$\text{sessioner}^{1}\left(p_1[i]\right) = \text{sessioner}^{1}\left(q_1[i]\right)$$

In the above definition of DAP the injective correspondence can be instead by non-injective correspondence.

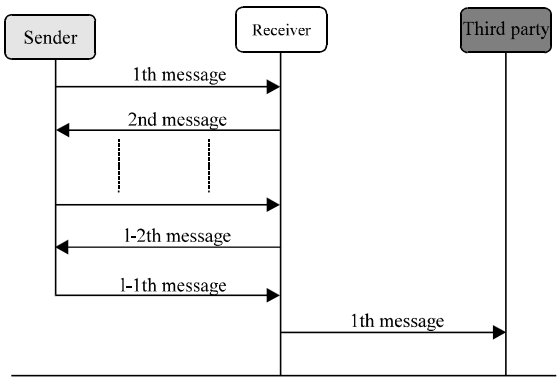

Fig. 6: The model of DAP

In deniable authentication protocol DAP, the first several messages between Sender and Receiver are executed, then the last message are send to Third party the by the Receiver. Generally the deniable authentication protocol consists of two sub protocols. The first sub protocol involved the Sender and Receiver. The second sub protocol called verification protocol run between the Third party and the Receive and it concludes one or two message from the Receive or/and the Sender according to the different model.

We assume that DAP consists of odd number of rounds l, so that the 1th message and l-1th message of DAP is from the Sender to the Receiver. The lth message is from the Receiver to the Third party. Here we only consider the only one message from the Receiver to the Third party. Figure 6 describes the model of DAP.

Init process Process is an initialization process which responsible for generating the relative parameters for Sender and Receiver, for example, the random numbers, public and private keysand so on. Sender process and Receiver process do not contain replication. Sender process process stores the messages of the deniable authentication protocol in variables:

$$x_1, x_2, \cdots, x_l$$

Receiver process process stores the message from Sender process process in variable:

$$z_1, z_2, \cdots, z_l$$

and stores the message to Third party process in variable p1. Third party process process stores it in variable q1.The Sender process process is invoked by a receiving message or external request which contains the identity Receiver of Receiver process process with Sender is supposed to run a session. After that the process

Receiver process is invoked by a receiving message or external request which contains the identity Third party of process Third party process with Third party is supposed to run a session. We designate by Sender process′ the copy of Receiver process of index isender = i and Receiver process′ by the copy Receiver process of ireceiver = i of and by Third party process′ the copy of Receiver process′ of ithird party = i. The l-1 th message is sent by Sender process process. The Receiver process′ process is assumed that it send the additional message one message including either:

$$1OK_{Receiver}(Sender)$$

the message of DAP protocol, where Receiver is the identity of its intended principle. We say that Receiver process′ finished with Sender when it sends:

$$OK_{Receiver}(Sender)$$

in the additional message two message.

Here we use the idea of the session function sessioner based on Blanchet (2007a, b) to match the conversion in the deniable authentication protocol. The session function is chosen to contain all messages of the protocol, except messages that are sent to or received from a trust third server. Messages that are just forwarded without checking (those can be changed by the adversary) and signatures when the security definition of signatures allows an adversary to forge a new signature for a message that has already been signed. A sessioner s a function of messages in the protocol:

$$sessioner(x_1, x_2, \cdots, x_l)$$

is typically a subsequence of the whole sequence messages:

$$x_1, x_2, \cdots, x_l$$

We also define a partial session function:

$$sessioner'(x_1, x_2, \cdots, x_{l-1})$$

useful since the lth message is not available to Receiver when Sender accepts. At the same time we also define a partial session function Sender″ (θ) useful since the message is not available to Third party when Receiver sends. We require that:

$$sessioner(x_1, x_2, \cdots, x_l) = sessioner(z_1, z_2, \cdots, z_l)$$

implies:

$$sessioner(x_1, x_2, \cdots, x_{l-1}) = sessioner(z_1, z_2, \cdots, z_{l-1})$$

We say that Sender process and Receiver process′ are intended principle when they have the same session function: and and are intended principles when they have the same session function:

$$sessioner(x_1[i], x_2[i], \cdots, x_l[i]) = sessioner(z_1[i], z_2[i], \cdots, z_l[i])$$

The condition one describes the communications between Sender and Receiver without adversary. It deal with the Receiver authenticates Sender. The condition two and three describe that the Sender has a distinct session with Receiver and the Receiver has the same session with Sender with overwhelming probability.

The condition four describes the communications between Receiver and Third party without adversary. It deal with the Third party authenticates Receiver. The condition five describes that the Receiver has a distinct session with intended principle Third party and the Third party has the same session with Receiver with overwhelming probability.

Definition DAP with events: If DAP′ satisfies the condition one and four in definition DAP and DAP′ satisfies the correspondence:

$$\begin{aligned} &inj\text{-}event(whole_{Sender}(Receiver, x)) \Rightarrow \\ &inj\text{-}event(whole_{Receiverr}(Sender, x)) \end{aligned} \quad (1)$$

$$\begin{aligned} &inj\text{-}event(whole_{Thirdparty}(Receiver, x)) \Rightarrow \\ &inj\text{-}event(whole_{Thirdparty}(Sender, x)) \end{aligned} \quad (2)$$

with public variables:

$$V = \varnothing$$

then DAP is a secure deniable authentication protocol with session functions (sessionid and sessionid′).

DAP′ process can be obtained from DAP by adding the following events in Table 1 and Fig. 7:

Formalizing strong deniability: Deniability consists of strong deniability and weak deniability. The purpose of strong deniability is to protect the privacy of Receiver.

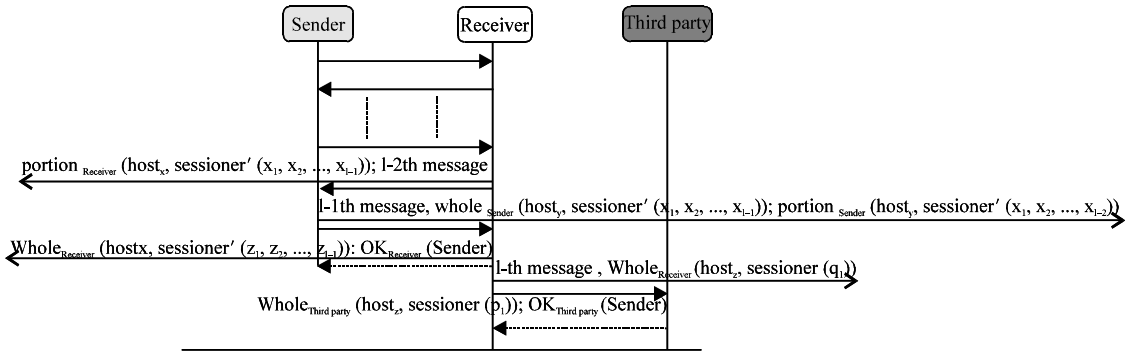

Fig. 7: The event in DAP. "→" means the sequence of output in the message in DAP

Table 1: The table: Event in DAP

Contents	Position
$\text{portion}_{\text{Receiver}}\left(\text{host}_x, \text{sessioner}'\left(z_1, z_2, \cdots, z_{l-2}\right)\right)$	Receiver sends l-2th message
$\text{portion}_{\text{Sender}}\left(\text{host}_y, \text{sessioner}'\left(x_1, x_2, \cdots, x_{l-2}\right)\right)$	Sender sends l-1th message, OKSender (hosty)
$\text{whole}_{\text{Sender}}\left(\text{host}_y, \text{sessioner}'\left(x_1, x_2, \cdots, x_{l-1}\right)\right)$	
$\text{whole}_{\text{Receiver}}\left(\text{host}_x, \text{sessioner}'\left(z_1, z_2, \cdots, z_{l-1}\right)\right)$	Receiver sends OKReceiver (Sender) in additional message one
message $\text{whole}_{\text{Receiver}}\left(\text{host}_z, \text{sessioner}\left(q_1\right)\right)$	Receiver sends lth message, OKReceive (hostz)
$\text{whole}_{\text{Thirdparty}}\left(\text{host}_z, \text{sessioner}\left(p_1\right)\right)$	Third party sends OKThird party (Sender) in additional two message

After execution of the deniable authentication protocol the Sender can deny to have ever authenticated anything to receiver. If the prover (receiver or the any other party) wants to prove that the Sender have authenticated messages to Receiver, they must provide all the relevant evidence. The sender can provide his secret information to the Third party.

A adversary model in strong deniability: When discussing the strong deniability, in addition the adversary has the ability in previous section, we always also suppose that the sender and the receiver cooperate with the judge or the prover or the any other party, which means that the sender and the receiver provide all the transcripts of the message in the deniable authentication protocol to them.

Definition of strong deniability: If DAP′ satisfies the condition one and four in definition DAP and DAP′ satisfies the correspondence:

$$\text{inj-event}\left(\text{whole}_{\text{Sender}}\left(\text{Receiver}, x\right)\right) \Rightarrow \text{inj-event}\left(\text{whole}_{\text{Receiverr}}\left(\text{Sender}, x\right)\right)$$

and:

$$\text{inj-event}\left(\text{whole}_{\text{Thirdparty}}\left(\text{Receiver}, x\right)\right) \Rightarrow \text{inj-event}\left(\text{whole}_{\text{Thirdparty}}\left(\text{Sender}, x\right)\right)$$

with public variables:

$$V = \varnothing$$

then DAP is a secure deniable authentication protocol with session functions (sessionid and sessionid′) in a adversary model in strong deniability. In the above definition of DAP the injective correspondence can be instead by non-injective correspondence.

Formalizing weak deniability: The purpose of weak deniability is to protect the privacy of Sender. After execution of the deniable authentication protocol the Receiver can prove to have spoken to Sender the but not the content of what the authenticated in a way that the Receiver can not convince a third party that such authentication. If the receiver want to prove that the sender have authenticated messages to receiver, he must provide the evidence related to the thing.

An adversary model in weak deniability: When discussing the weak deniability, in addition the

adversary has the ability in previous section; we always suppose that only the receiver generates the evidence that the sender have authenticated messages to receiver. The Receiver can not get the secret information of the sender, for example the private key of the Sender. The Receiver can provide his secret information to the Third party.

Definition of weak deniability: If DAP′ satisfies the condition one in definition DAP and DAP′ satisfies the correspondence:

$$\text{inj-event}\left(\text{whole}_{\text{Sender}}\left(\text{Receiver}, x\right)\right) \Rightarrow \text{inj-event}\left(\text{whole}_{\text{Receiverr}}\left(\text{Sender}, x\right)\right)$$

and:

$$\text{inj-event}\left(\text{whole}_{\text{Thirdparty}}\left(\text{Receiver}, x\right)\right) \Rightarrow \text{inj-event}\left(\text{whole}_{\text{Thirdparty}}\left(\text{Sender}, x\right)\right)$$

with public variables:

$$V = \varnothing$$

then is a secure deniable authentication protocol with session functions (sessionid and sessionid′) in a adversary model in weak deniability. In the above definition of DAP the injective correspondence can be instead by non-injective correspondence.

CONCLUSION

During the past few decades deniable authentication protocol has been studied. A lot of deniable authentication protocols have been proposed which claimed that have the security properties, for example, authentication; strong deniability; weak deniability; security of forgery attack; security of impersonate attack; security of compromising session secret attack; security of man-in-the-middle attack and so on. In order to verify the security properties of security protocol including deniable authentication protocols and increase the confidence of the people, two approaches have been developed for analyzing security protocols form the beginning of the 1980s. One approach relies on a symbolic model of protocol executions in which cryptographic primitives are treated as black boxes. The other approach relies on a computational model that base issues of complexity and probability. This approach use a strong notion of security, guaranteed against all probabilistic polynomial-time attacks. The computation approach can be more realistic. To our best knowledge, this security properties and deniable authentication protocols are analyzed with informal method or with symbolic method by hand which depends on experts' knowledge and skill and is prone to make mistakes.

So analysis of security properties and deniable authentication protocols with automatic tool in symbolic model or computational model plays an important role in security protocol world and is a significant work. Especially analysis of security properties and deniable authentication protocols with automatic tool in computational model is a changeling issue.

Recently owning to the introduction of Blanchet calculus and a powerful mechanized tool CryptoVerif we use Blanchet calculus in the computational model to analyze the typical deniable authentication protocols with CryptoVerif.

In this study, firstly the state-of-art of deniable authentication protocol and the proof including in symbolic model and in computational model are presented. We find that security properties and deniable authentication protocols are analyzed with informal method or with symbolic method by hand.

Then the term, process and correspondence assertion in Blanchet calculus is used to model the security properties including strong deniability and weak deniability and deniable authentication protocol. Finally we propose the first mechanized framework of deniable authentication protocols in computational model with active adversary. The strong deniability and weak deniability are expressed by injective correspondence. This mechanized framework can be used to automatically analyze the security properties including strong deniability and weak deniability of interactive deniable authentication protocols and non-interactive deniable authentication protocols with CryptoVerif.

Further studies concentrate on application on several typical deniable authentication protocols, for example, Fan *et al.* (2002) interactive deniable authentication protocol, Meng (2009a) non-interactive deniable authentication protocol, with CryptoVerif. At the same time we also work on formal model of the security of forgery attack, security of impersonate attack, security of compromising session secret attack, security of man-in-the-middle attack.

ACKNOWLEDGMENTS

This study was supported in part by Natural Science Foundation of South-Center University for Nationalities under the grants No: YZZ06026, titled "Research on the internet voting protocols with receipt-freeness", conducted in Wuhan, China from 06/11/2006 to 20/11/2009.

REFERENCES

Aumann, Y. and M. Rabin, 1998. Efficient deniable authentication of long messages. Proceedings of the International Conference on Theoretical Computer Science in Honor of Professor Manuel Blum's 60th Birthday, 1998. http://www.cs.cityu. edu. hk/dept/video. html.

Bellare, M., R. Canetti and H. Krawczyk, 1998. A modular approach to the design and analysis of authentication and key exchange protocols. Proceedings of the 30th Annual Symposium on the Theory of Computing, (ASTC'98), Dallas, Texas, United States, pp: 419-428.

Bhargavan, K., R. Corin and C. Fournet, 2007. Crypto-verifying protocol implementations in ML. Proceedings of the Workshop on Formal and Computational Cryptography-FCC 2007.

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. and D. Pointcheval, 2006. Automated security proofs with sequences of games. Proceedings of the 27th IEEE Symposium on Security, Aug. 06, LNCS, Santa Barbara, CA, Springer Verlag, pp: 537-554.

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.

Blanchet, B., 2005. A computationally sound mechanized prover for security protocols. Cryptology ePrintArchive, Report 2005/401, Nov. 2005. http://eprint.iacr.org/2005/401.

Blanchet, B., 2006. A computationally sound mechanized prover for security protocols. Proceedings of IEEE Symposium on Security and Privacy, May 21-24, Oakland, California, pp: 150-154.

Blanchet, B., 2007a. A computationally sound mechanized prover for security protocols. IEEE Trans. Dependable Secure Comput., 5: 193-207.

Blanchet, B., 2007b. Computationally sound mechanized proofs of correspondence assertions. Proceedings of 20th IEEE Computer Security Foundations Symposium, July 6-8, Venice, Italy, pp: 97-111.

Blanchet, B., A.D. Jaggard, A. Scedrov and J. Tsay, 2008. Computationally sound mechanized proofs for basic and public-key kerberos. Proceedings of the ACM Symposium on Information, Computer and Communications Security, March 2008, Tokyo, Japan, pp: 87-99.

Canetti, R. and H. Krawczyk, 2001. Analysis of key-exchange protocols and their use for building secure channels. Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology, May 06, Springer-Verlag, London, UK., pp: 453-474.

Canetti, R. and H. Krawczyk, 2002. Universally composable notions of key exchange and secure channels. Adv. Cryptol., 2332: 337-351.

Deng, X., C.H. Lee and H. Zhu, 2001. Deniable authentication protocols. IEE Proc. Comput. Digital Techniques, 148: 101-104.

Dwork, C., M. Naor and A. Sahai, 1998. Concurrent zero-knowledge. Proceedings of the 30th Annual ACM Symposium on Theory of Computing, 1998, USA., pp: 409-418.

Fan, L., C.X. Xu and J.H. Li, 2002. Deniable authentication protocol based on Diffie–Hellman algorithm. Elect. Lett., 38: 705-706.

Feng, T. and J.F. Ma, 2007. Universally composable security concurrent deniable authentication based on witness indistinguishable. J. Software, 18: 2871-2881.

Han, S., W.Q. Liu and E. Chang, 2005. Deniable Authentication protocol resisting man-in-the-middle attack. Proc. World Acad. Sci. Eng. Technol., 3: 161-164.

Jaggard, A.D., A. Scedrov and J. Tsay, 2007. Computationally sound mechanized proof of PKINIT for kerberos. Proceedings of the Workshop on Formal and Computational Cryptography - FCC 2007. http://dimacs.rutgers.edu/~adj/Research/papers/jst07fcc.pdf

Laud, P., 2005. Secrecy types for a simulatable cryptographic library. Proceedings of the 12th ACM Conference on Computer and Communications Security, Nov. 07-11, ACM, New York, pp: 26-35.

Lee, W.B., C.C. Wu and W.J. Tsaur, 2007. A novel deniable authentication protocol using generalized ElGamal signature scheme. Inform. Sci., 177: 1376-1381.

Lincoln, P., J. Mitchell, M. Mitchell and A. Scedrov, 1998. A probabilistic poly-time framework for protocol analysis. Proceedings of the 5th ACM Conference on Computer and Communications Security, Nov. 02-05, San Francisco, California, United States, pp: 112-121.

Lincoln, P., J.C. Mitchell, M. Mitchell and A. Scedrov, 1999. Probabilistic polynomial-time equivalence and security analysis. Proc. World Cong. Formal Meth. Dev. Comput. Syst., 1: 776-793.

Lu, R. and Z. Cao, 2005a. A new deniable authentication protocol from bilinear pairings. Applied Math. Comput., 168: 954-961.

Lu, R. and Z. Cao, 2005b. Non-interactive deniable authentication protocol based on factoring. Comput. Standards Interfaces, 27: 401-405.

Meng, B., 2009a. 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., 2009b. Formalizing deniability. Inform. Technol. J., 8: 625-642.

Micciancio, D. and B. Warinschi, 2004. Soundness of formal encryption in the presence of active adversaries. Theory Cryptography (LNCS), 2951: 133-151.

Qian, H.F., Z.F. Cao, L.C. Wang and Q.S. Xue, 2005. Efficient non-interactive deniable authentication protocols. Proceedings of the 5th International Conference on Computer and Information Technology, Sept. 21-23, IEEE Computer Society Washington, DC. USA., pp: 673-679.

Raimondo, M.D. and R. Gennaro, 2005. New approaches for deniable authentication. Proceedings of the 12th ACM Conference on Computer and Communications Security, Nov. 7-11, ACM Press, New York, pp: 112-121.

Shao, Z., 2004. Efficient deniable authentication protocol based on generalized ElGamal signature scheme. Comput. Standards Interfaces, 26: 449-454.

Shi, Y. and J. Li, 2005. Identity-based deniable authentication protocol. Elect. Lett., 41: 241-242.