Computer Assisted Proof of Resistance of Denial of Service Attacks in Security Protocols Based on Events with CryptoVerif in Computational Model

1 Bo Meng, 2 Wei Chen

1, First Author School of Computer, South-Center University for Nationalities, mengscuec@gmail.com
*1, Corresponding Author School of Computer, South-Center University for Nationalities, mengscuec@gmail.com
2 School of Computer, South-Center University for Nationalities, chenw2011@hotmail.com

Abstract

Owing to the huge damage and difficulty of prevention of denial of service attacks in security protocols, people pay serious attentions on analysis, verification and prevention of denial of service attacks. Formal method in computational model is a powerful tool used to analyze and prove securities in security protocols, network and distributed system. But until now in computational model there does not exist an automatic method of proof of resistance of denial of service attacks. In this paper from the view of protocol state we initially present an automatic method of proof of resistance of denial of service attacks with CryptoVerif in computational model. Therefore Blanchet calculus is extended from adversary context, output process and the model of private channel, and then from the view of protocol state, the first automatic method of proof of resistance of denial of service attacks based on events in extended Blanchet calculus is proposed. Finally resistance of denial of service attacks in IEEE 802.11 i four-way handshake protocol is analyzed with CryptoVerif. The results we obtained are that it is not resistance of denial of service attacks. At the same time a new denial of service attack is found by us. Then the methods against denial of service attacks in IEEE 802.11 i four-way handshake protocol are proposed.

Keywords: automatic verification, protocol state, computational model, availability

1. Introduction

With the development of Internet and information technology, electronic government has got serious attention from government, enterprise and academic world. Owning to advantages of remote internet voting, it plays an important role in electronic government. In order to increase confidence of the voters in remote internet voting system, many researchers have focused on design and verification on secure remote internet voting systems and protocols. Remote internet voting protocol is a key part of internet voting system. So how to develop and verify a practical secure internet voting protocol is a challenging issue.

Owing to the huge damage and hardiness of prevention of denial of service attacks in security protocol, network and distributed system, people pay serious attentions on analysis, verification and prevention of denial of service attacks. This kind of attack aims at rendering a network or a system incapable of providing normal service by targeting the network, bandwidth or connectivity. Denial of service attacks is simple and effective, for example, the adversary can create many bogus messages and send it to target of attack. That make the target of attack can not provide normal service for legitimate user owning to process big bogus messages. At the same time it is not easy to find the adversary and adversary can mount another type attacks through denial of service attacks, for example, man-in-the-middle attack.

In order to prevent denial of service attacks the first step is to analyze and prove resistance of denial of service attacks in protocol, network and distributed system with formal method and give the confidence of people in its security. There are two models can be used: one is symbolic model in which...
cryptographic primitives are ideally abstracted as black boxes, the other is computational model based on complexity and probability theory. The last model is more realistic.

In symbolic model there are mainly three formal frameworks in resistance of denial of service attacks. One is Yu-Gligor model [1] based on user agreement. The core of framework is based on access control policy. But it can not deal with denial of service attacks executed before authentication between sender and receiver in protocol, for example, SYN floods attacks. At the same time it does not get the help of the automated tools. The second one is Meadows’s cost-based model [2] which built on fail-stop protocol. People pay much attention to it. The third one is Meng-Huang model [3] that is the first automatic method of resistance of denial of service attacks in symbolic model with ProVerif.

To our best knowledge in computational model analysis model of resistance of denial of service attacks has not been proposed.

Recently Blanchet [4] propose a probabilistic polynomial calculus based on computational model. In this calculus, messages are bitstrings and cryptographic primitives are functions operating on bitstrings. The 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. At the same time he develops a mechanized tool CryptoVerif which is the only automatic tool with computational model until now. It can directly prove security properties of cryptographic protocols in the computational model. In a recent case study, CryptoVerif is used to verify many crypto cryptographic primitives and security protocols.

Owing to many denial of services attacks are launched by protocol state, for example, the attacker sends fake requests to the server, causing the server to exhausts all of its resources to processing and maintaining the malicious requests state and to can not accept requests from legitimate clients; the attacker alter the protocol state and make the protocol states inconsistence and launch denial of service attacks, hence in this paper resistance of denial of service attacks is modeled from the view of protocol state. In order to model protocol state and resistance of denial of service attacks, Blanchet calculus is extended from three aspects: adversary context, output process and model of private channel, and then from the view of protocol state, the first automatic method of proof of resistance of denial of service attacks based on events in computational model is proposed. The main contributions of this paper are summarized as follows:

- Review the formal models of resistance of denial of service attacks in security protocols. Until now analysis model of resistance of denial of service attacks based on computational model has not been proposed.
- In order to model the protocol state and resistance of denial of service attacks, Blanchet calculus is extended from three aspects: one is the adversary context, the other is output process, the third is the model of private channel.
- Based on extended Blanchet calculus proposed by us, from the view of protocol state, the first automatic method of resistance of denial of service attacks based on events in computational model is proposed.
- Resistance of denial of service attacks in IEEE 802.11 i four-way handshake protocol [5] is analyzed with CryptoVerif based on method proposed by us. The results we obtained are that IEEE 802.11 i four-way handshake protocol is not. Simultaneously a new denial of service attack is found by us. Then the methods to prevent resistance of denial of service attacks in IEEE 802.11 i four-way handshake protocol are proposed.

2. Extended Blanchet calculus

Extended Blanchet calculus is based on Blanchet calculus which is a probabilistic polynomial calculus and has been carefully designed to make the automated proof security protocols. In Blanchet calculus, messages are bitstrings and cryptographic primitives are functions operating on bitstrings. 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 and can get the help of mechanized proof CryptoVerif.

In order to model the protocol state and resistance of denial of service attacks, we extend Blanchet calculus from three aspects: one is the adversary context, the other is output process, and the third is
model of private channel. The semantic of extended Blanchet calculus is same to the one of Blanchet calculus and also can get the help of CryptoVerif.

**Adversary contexts**

The adversary is presented by an evaluation context \( C \). An evaluation context is a process with a hole, of one of the following forms: a hole \( \_ \_ \_ \_ \), a process in parallel with an evaluation context \( Q' \), or a restriction \( new\text{Channel}C \)., 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 \( Q \). In extended Blanchet calculus, according to abilities of adversary the contexts of adversary are classified into two contexts: one is real context, the other is ideal context. Real context is formalized as \( C[\pi(x).P], C[u(x).Q] \). Intuitively, real context is insecure environments where the adversary is in computational model. Ideal context is formalized as \( C[\pi(\_).N.P], C[u(\_).N.Q] \). Intuitively ideal context is secure environments.

**Extend output process**

In extended Blanchet calculus, output processes \( P \) in Figure 1 output a message on a channel after executing some internal computations.

![Output process](image)

**Figure 1. Output process**

The output process \( new \ x[i_1,\ldots,i_n] ; T ; P \) chooses a new random number uniformly in \( I(T) \), stores it in \( x[i_1,\ldots,i_n] \), and executes \( P \). Function symbols represent deterministic functions, so all random numbers must be chosen by \( new \ x[i_1,\ldots,i_n] ; T \). Deterministic functions make automatic syntactic manipulations easier: it can be duplicated by a term without changing its value. The process \( let \ x[i_1,\ldots,i_n] ; T = M \ in \ P \) stores the bitstring value of \( M \) in \( x[i_1,\ldots,i_n] \) and executes \( P \). The conditional construct \( if \ defined \{M_1,\ldots,M_l\} \wedge M \ then \ P \ else \ P' \) runs that if \( defined \{M_1,\ldots,M_l\} \wedge M \) is true, executes \( P \), otherwise executes \( P' \). The conditional construct \( if \ defined \{M_1,\ldots,M_l\} \wedge M \ then \ P \ else \ C \{existdos(M,N) \} ; P' \) runs that if \( defined \{M_1,\ldots,M_l\} \wedge M \) is true, executes \( P \), otherwise executes \( C \{existdos(M,N) \} ; P' \) in idea context. The model of private channel

The model of private channel is not supported in CryptoVerif, so here we give the method of model of private channel in Blanchet calculus which can 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 can get the information in a way but the adversary does not access the information.

For example: 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.

If there is a single session of A, we can simply write in Figure 2:

\[
\text{Figure 2. A single session}
\]

\[
\text{in A: new n : nonce}
\]
\[
\text{in B: find suchthat defined(n) then ...n...}
\]

If there are several sessions of A, a solution is to have public session id in Figure 3:

\[
\text{Figure 3. Several sessions}
\]

\[
\text{type sessionid [large].}
\]
\[
\text{in A: !N}
\]
\[
\text{new n: nonce;}
\]
\[
\text{new id: sessionid;}
\]
\[
\text{out(cA, id)}
\]
\[
\text{in B: in(cB, idA: sessionid);}
\]
\[
\text{find j<N suchthat defined(n[j],id[j]) && idA = id[j] then ... n[j]...}
\]

3. Definitions of resistance of denial of service attacks

\( P \) is an annotated Alice-and-bob specification in protocol, \( B \) is resistance of denial of service attacks if and only if set of association \( \omega \) between any message \( M_i \) and \( M_j \) in set \( \text{Recv}(B) \):

1. \( \omega \) is null set \emptyset
2. Any data items in \( \omega \) are authenticated.

Where \( \text{Recv}(B) \) is set where data items are in operations that are ordered in casually precedes in \( \text{act}_i(B)[M_i, O_i, \ldots, O_i] \), where \( i, j \in [1,n] \) and \( i < j \).

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

4. Automated proof of resistance of denial of service attacks based on events in CryptoVerif

Applying the extended Blanchet calculus the protocol can be modeled as an annotated Alice-and-Bob specification. We assume that the protocol exchanges \( 2n \) messages between principle Alice and Bob in a run. Principle Alice sends \( n \) messages \( M_i \), where \( i \in [1,n] \), and receives \( n \) messages \( M_i' \), where \( i \in [1,n] \). Principle Bob receives \( n \) messages \( M_i \), where \( i \in [1,n] \) and sends \( n \) messages \( M_i' \), where \( i \in [1,n] \). Protocol process \( PP = \text{Initprocess}\(\text{\{} Alice, Bob \text{\}} \) is a process and consists of parallel composition of initialization process \(\text{Initprocess} \), initiator process \( Alice \) and responder process \( Bob \).
In order to use CryptoVerif to automatically prove resistance of denial of service attacks of Bob, the any messages $M_i$, where $i \in [1, n-1]$, is modeled with the extended Blanchet calculus. If query event existsDos$(M, N)$ is true, adversary can launch a denial of service attack by attack of message $M_i$, where $i \in [1, n-1]$.

![Figure 4. The model of messages $M_i$, $i \in [1, n-1]$](image)

The method is used to model the messages $M_i$ where $i \in [1, n-1]$ in Figure 4. The message $M_i$ is exchanged and processed in real context. The messages $M_j$ where $j \in [1, n-1], j \neq i$ and $M_j'$ where $j \in [1, n-1]$ are exchanged and processed in idea context. Protocol process $PP$ is $PP = \text{Initprocess}[\{Alice\ \{Bob\}]$, where $c$ is public channel, $c_j$, where $j \in [1, n-1] \cap j \neq i$, are private channels used to receive messages $M_i$, where $i \in [1, n-1]$ and $j \in [2, n-1] \cap j \neq i$.

Alice$_{c}$ $\xrightarrow{\text{C}[\{c\}]\{M\}}$ Alice$_{c1}$, where $c \notin \tilde{n}, c \in \tilde{n}$; Bob$_{c}$ $\xrightarrow{\text{C}[\{x\}]\{m\}}$ Bob$_{c1}$, where $c \notin \tilde{n}$; Alice$_{c_j}$ $\xrightarrow{\text{C}[\{M\}]\{c\}}$ Alice$_{c_{j1}}$, where $c_j \in \tilde{n}, j \in [1, n] \cap j \neq i$; Bob$_{c_j}$ $\xrightarrow{\text{C}[\{m\}]\{c\}}$ Bob$_{c_{j1}}$, where $c_j \in \tilde{n}, j \in [1, n] \cap j \neq i$. If query event existsDos$(M, N)$ is true, the adversary can launch a denial of service attacks by attacks of message $M_i$, where $i \in [1, n-1]$.

**Theorem:** resistance of denial of service attacks in computational model

Responder Bob in protocol process $PP$ is resistance of denial of service attacks if and only if the formal model of all messages $M_i, i \in [1, n-1]$ received by principle Bob, in $PP$ all query event existsDos$(M, N)$ are not true, in other words, there does not processes $P^*, P^*$ and attacker process $Attacker$ to cause $\langle PP \mid Attacker \rangle \xrightarrow{\text{existsDos}(M, N)} P^* | \text{c} \notin \tilde{n}, \text{S} \notin \tilde{n}$.

**Proof:**

If Bob has the ability of resistance of denial of service attacks, according to definition 6 of resistance of denial of service attacks, $\forall M_i, \forall M_j \in \text{Recv}(Bob)$, $i, j \in [1, n]$, $M_i$ casually precedes $M_j$, verification operation $v$ is if $M = N$ then $P$ else $C[\text{existsDos}(M, N)]Q \in \text{act}(Bob[M])$, then for the formal model of message $M_i$:

1. $\omega = \emptyset$, in other words, the value of $M = N$ in the annotated Alice-and-bob specification in protocol is not related to message $M_i$, then $M_i$ is exchanged in idea context or real context, hence the value of $M = N$ is always true, protocol process $PP$ $\xrightarrow{\text{existsDos}(M, N)} P$.
2. $\omega = \{\tilde{m}\}$, $\tilde{m}$ is authenticated data items, in other words attacker can not alter $\tilde{m}$, hence the
value of $M = N$ is always true whatever $M_i$ is exchanged in idea context or real context, protocol process $PP \xrightarrow{*} P$.

Hence for the formal model of all messages $M_i, i \in [1, n - 1]$ received by principle $Bob$, in $PP$ all query event $\text{existsdos}(M, N)$ are not true.

For the model of all messages $M_i, i \in [1, n - 1]$ received by principle $Bob$, if in $PP$ all query event $\text{existsdos}(M, N)$ failed, then it is can included that there exist processes $P'$, $P''$ and attacker process $Attacker$ to cause $\{PP | Attacker\} PP \xrightarrow{*} P \text{existsdos}(M, N), P^* \not\in \tilde{n}$. Hence attacker $Attacker$ can make that the value of $M = N$ is always false by altering the message which is casually precedes $M_j$ and is associated to $v$. Then set of association $\omega$ has the data items $\tilde{m}$ which are not authenticated, so responder $Bob$ in protocol process $PP$ is not resistance of denial of services attacks.

According to the theorem in $PP$ query event $\text{existsdos}(M, N)$ is true, and then attacker can construct a denial of service attack by altering the message $M_j$ which make the receiver can not find, without influence on other messages exchanges in protocol.

Hence the extended Blanchet calculus can be used to model resistance of denial of service attacks, and then based on the proposed theorem, apply CryptoVerif to automatically prove the resistance of denial of service attack.

5. Mechanized proof tool CryptoVerif

In this section we give a brief overview of the mechanized prover CryptoVerif. 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 prove secrecy properties and events that can be executed only with negligible probability; also it can handle various cryptographic primitives. CryptoVerif works for $N$ sessions with an active adversary. In a recent case study, CryptoVerif is used to verify many security protocols [6,11,12,13].

Games approach

CryptoVerif prove security using the sequence-of-games approach. Figure 5 describes the ideal of security properties proof with sequences of games.

![Figure 5. The ideal of security properties proof with sequences of games](image)

1. 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.
2. Generally makes transformation between successive games based on one of the methods: indistinguishability, failure events and bridging steps and then generate new attack game.
3. Evaluate the change between two consecutive attack games.
4. 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 Figure 6, 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.

![Figure 6. The ideal of security properties automatic proof in CryptoVerif](image)

**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 on 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 [7]. CryptoVerif takes as input a script in which cryptographic assumptions are introduced through type and function declarations, equations, inequations, and game-based equivalences.

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.

6. Case: IEEE 802.11 four-way handshake protocol

Wireless Local Area Networks plays an important role in our digital society. IEEE 802.11 standard aims to provide the secure communication channel between principles in the 802.11i protocol. In addition to introducing key management and establishment, it also defines encryption and authentication improvements. There are three parties: the supplicant, the authenticator and the authentication server involved in the 802.11i protocol. The complete process of an 802.11i authentication consists of handshakes between supplicant and authenticator, between authenticator and authentication server, and between supplicant and authentication server.
He and Mitchell [8] analyze the 4-Way Handshake protocol using a finite-state verification tool using Murϕ and find a denial of service attack. They also proposed three repairs based on various considerations. Bicakcia and Tavli [9] present a systematic survey of denial of service attacks, which exploits MAC and physical layer vulnerabilities of 802.11 networks.

| message1: Alice → Bob : Anonce, \(m_1\) |
| message2: Bob → Alice : Snonce, \(m_2\), MIC_{s} |
| message3: Alice → Bob : Anonce, \(m_3\), MIC_{s} |
| message4: Bob → Alice : \(m_4\), MIC_{s} |

**Figure 7.** The simplified four-way handshake protocol

Where *Alice* represents the authenticator; *Bob* represents supplicant; Anonce and Snonce are the random number generated by *Alice* and *Bob*, respectively; Anonce and Snonce are used to generate the PTK that divided into three keys: KCK (Key Confirmation Key) which is used by the EAPOL-key exchanges to provided data origin authenticity., KEK (Key Encryption Key) which is used by the EAPOL-key exchanges to provide for confidentiality and TK (Temporary Key) which is used by the data-confidentiality protocols. \(m_1, m_2, m_3, m_4\) includes the key replay counter \(\text{Replay Counter}\). \(\text{MIC}()\) represent Message Integrity Code function; \(\text{MIC}_s = \text{MIC}(\text{Snonce}, m_2)\), \(\text{MIC}_i = \text{MIC}(m_1)\).

Four-way handshake protocol consists of four messages in Figure 7:

1. **Authenticator *Alice*** first sends message message1 to supplicant *Bob*. message1 includes random number Anonce and some secret keying material.
2. **Supplicant *Bob*** receives the message message1, then it check key replay counter \(\text{Replay Counter}\), if the result is true then it creates random number Snonce. He also applies the Pseudo Random Functions (PRF) to generate the PTK. PRF accepts Anonce, Snonce, MSK generated after authentication between supplicant and authentication server in IEEE 802.1X, the supplicant's MAC address and the authenticator's MAC address as its input. Supplicant *Bob* produces \(\text{MIC}_s = \text{MIC}(\text{Snonce}, m_2)\), then send message message2 to authenticator *Alice*. The supplicant *Bob* also sends the security parameters that it used during association. The entire message gets an authentication check using the KCK from the pairwise key hierarchy.
3. **Authenticator *Alice*** receives message message2 and computes PTK. At the same time he also checks \(\text{MIC}_s = \text{MIC}(\text{Snonce}, m_2)\) in message message2. If the result is true, then he generates message message3 and sends it to supplicant *Bob*. otherwise message message2 are discarded. Message message3 includes \(\text{MIC}_i = \text{MIC}(\text{Anonce}, m_1)\), Anonce and other information. the entire message gets an authentication check, which allows the supplicant to verify that the information is valid.
4. **Supplicant *Bob*** receives message message3 and checks \(\text{MIC}_i = \text{MIC}(\text{Anonce}, m_1)\). If the result is true then he generates message message4 and sent it to authenticator *Alice* and configure PTK. Authenticator *Alice* receives message message4 and checks \(\text{MIC}_i = \text{MIC}(m_1)\). If the result is true then he configures PTK. At this time the temporal keys are now in place to be used by the data-confidentiality protocols.

**7. Automatic proof of four-way handshake protocol with CryptoVerif**

Based on the proposed method of automatic proof of resistance of denial of service attack, the extended Blanchet calculus is used to model message received by supplicant *Bob*. Then it is translated the input language of CryptoVerif. Owning to the space limitation we only gives the results of analysis of CryptoVerif. The codes of four-way handshake protocol in inputs of CryptoVerif are in Figure 11 and 13. According to results of analysis of CryptoVerif, we find two denial of service attacks: denial of
service attack one is found by us with our proposed method.

\[
\text{fun PRF(nonce, nonce, key): key [compos].}
\]

\[
\text{fun MIC(nonce, counter, key): macs [compos].}
\]

\[
\text{fun MIC2(counter, key): macs [compos].}
\]

channel c1, c2, c3, c4, c, start, finish.

query Anonce: nonce; Replay_Counter: counter; existdos(nonce, counter).

let processA =
in(c, ());
new Anonce_A : nonce;
new m1_A : counter;
out(c1, (Anonce_A,m1_A));
in(c2, (idB_A:sessionid)); (* Receiving M2*)
find j1 <= N suchthat
defined(idB_B[j1], Snonce_B[j1], m2_B[j1], MIC2_B[j1]) &&
(idB_A = idB_B[j1])
then
let Snonce_A = Snonce_B[j1] in
let m2_A = m2_B[j1] in
let MIC2_A = MIC2_B[j1] in
let PTK_A = PRF(Anonce_A, Snonce_A, MSK) in
new m3_A: counter;
new idA_A: sessionid;
let MIC3_A: macs = MIC(Anonce_A, m3_A, PTK_A) in
out(c3, (idA_A)); (*Sending session id*)

let processB =
in(c1, (Anonce_B, nonce, m2_B[counter])); (*Receiving M1*)
find j3 <= N suchthat
defined(idA_B[j3]) && (m1_B = m1_A[j3])
then
let Anonce_B_2 = Anonce_A[j4] in
let m3_B = m3_A[j4] in
let MIC3_B = MIC3_A[j4] in
if MIC3_B = MIC(Anonce_B_2, m3_B, PTK_B) then (*event*)
(out(c4, (idB_B)));
else
event existdos(Anonce_B, m3_B);
out(c4, (idB_B)). (*Sending sessionid*)

let new MSK: key; out(c, ());
((! N processA) | (! N processB))

**Figure 8.** The code of formal of resistance of denial of service attack one in four-way handshakes protocol.

According to the specification of four way handshake protocol the supplicant Bob verify the message message3 based on PTK, while the generation of PTK is based on Anonce and Snonce. The generation of Snonce is based on the verification of key replay counter Replay.Counter. If the adversary fake or replay Anonce or Replay.Counter, then the verification of message message3 will be failed, hence the denial of service attack is launched.
Based on the formal model of message $message_1$, the operations on supplicant $Bob$ deal with $message_1$ and $message_3$ are $\text{Recv}(Bob) \left[ \text{act}(Bob) \left[ M_i, O_i^1, \ldots, O_i^k, \right] \right] \text{act}(Bob) \left[ M_i, R_i^0, \ldots, R_i^k \right]$,
where $v$ is verification operation $\{\}$
$MIC_i = H \left( KCK \right) \left( \text{Anonce}_i, m_i \right) \quad KCK = \{ \text{Anonce}', \text{Snonce}, \text{MSK} \}$, $M_i = \{ \text{Anonce}_i, m_i \}$,
$M_i = \{ \text{Anonce}_i, m_i, MIC_i \}$, so the set of association in message $message_1$ and $message_3$ is $\omega = \{ \text{Anonce}' \}$. Owning to the $\text{Anonce}'$ is not authenticated in four way handshake protocol, we can get that four way handshake protocol is not denial of service attack according to the definition 6.

Denial of service attack one in Figure 8 and 9. Figure 8 shows that the query event $\text{existsdos}(x, y)$ can not be proved. In an run of the four-way handshake protocol, before the time supplicant $Bob$ receives message $message_3$ and after the time supplicant $Bob$ receive message $message_2$ the adversary impersonates authenticator $Alice$ then modifies the message $message_1$ and construct fake $\text{Anonce}'$ and $\text{Replay Counter}'$, and then generate message $message_2$ and send it to supplicant $Bob$, according to the specification on four way handshake protocol supplicant $Bob$ generates PTK’ once more and update his cache, at the time supplicant $Bob$ receive genius message $message_3$, thus the verification of $MIC_i = H \left( KCK \right) \left( \text{Anonce}_i, m_i \right)$ with PTK’, the result is false because PTK is different from the one in the authenticator. Thus this attack makes the PTK inconsistency. According to specification on four way handshake protocol, authenticator $Alice$ will again send the message $message_3$, but the verification of $MIC_i = H \left( KCK \right) \left( \text{Anonce}_i, m_i \right)$ is again false. After several times of this verification, authenticator $Alice$ and supplicant $Bob$ need mutually authenticated again. Hence it is a denial of service attack.

![Figure 9. The result of resistance of denial of service attack one in four-way handshakes protocol](image)

![Figure 10. Denial of service attack one](image)
Dehydration of service attack two in Figure 10 and 11. Figure 10 shows that query event

\texttt{existsdos(x,y)}

can be proved. In an run of the four-way handshake protocol, before the time supplicant Bob receives message message3 and after the time supplicant Bob message message2, the adversary impersonates authenticator Alice, then modifies the message message1 and construct a
fake Replay \_Counter', and then generate message message1 and send it to supplicant Bob, according to the specification on four way handshake protocol supplicant Bob check Replay \_Counter', but does not check that if the Anonce is replay or not generates a new Snonce and PTK' once more and update his cache, at the time supplicant Bob receives genius message message3, thus the verification of \( \text{MIC}_3 = \text{MIC}(\text{Anonce}, m_3) \) with PTK', the result is false because PTK is different from the one in the authenticator. Thus this attack makes the PTK inconsistency. According to specification on four way handshake protocol, authenticator Alice will again send the message message3, but the verification of \( \text{MIC}_3 = \text{MIC}(\text{Anonce}, m_3) \) is again false. After several times of this verification, authenticator Alice and supplicant Bob need mutually authenticated again. Hence it is a denial of service attacks.

**Figure 12.** The new resistance of denial of service attack

**Figure 13.** The new denial of service attack

In order to against this denial of service attack one, He and Mitchell [8] argue that supplicant Bob store the PTK and the TPTK (Temporary PTK) for each message message1. When it receives the message message1 it only updates TPTK. Only it receives the genius message message3 he updates PTK. But when the adversary sends many bogus message message1, supplicant Bob need to store many TPTK thus it make a resource exhaustion of denial of service attack.

Based on the previous analysis that it is make the messages message1 is authenticated is to protect four way handshake protocol against the two denial of service attacks. For example, sign messages message1 with private key of authenticator Alice, or encrypt the messages message1 with MSK, and so on. If we only protect it against denial of service two, then we need to let supplicant Bob to store the Anonce in a time and check that whether Anonce is replay or not before the next operation.

**8. Conclusion and future work**

Owning to the huge damage and hardness of prevention of denial of service attacks in security protocols, people pay serious attentions on analysis, verification and prevention of denial of service attacks. In order to prevent denial of service attacks the first step is to analyze and proof resistance of
denial of service attacks in protocol, network and distributed system with formal method and give the confidence of people in its security.

In order to model the protocol state and resistance of denial of service attacks in computational model, firstly Blanchet calculus is extended from three aspects: one is the adversary context, the other is output process, the third is the model of private channel, then from the view of protocol state, the first automatic method of resistance of denial of service attacks based on event in computational model is proposed. Finally the resistance of denial of service attacks in IEEE 802.11 i four-way handshake protocol is analyzed with CryptoVerif based on the formal method proposed by us. The results we obtained is that it is not. A new denial of service attack in IEEE 802.11 i four-way handshake protocol is found by us. Then the methods against resistance of denial of service attacks are proposed.

As future work, we plan to prove resistance of denial of service attacks in electronic commerce protocols.

9. Acknowledgement

This study was supported in part by Natural Science Foundation of The State Ethnic Affairs Commission of PRC under the grants No: 10ZN09. The paper is the extended full version [10]. We thanks Bruno Blanchet for his help on model of private channel in CryptoVerif and comments.

10. References