ISSN : 0974 - 7435 Volume 10 Issue 23

2014

BioTechnology
An Indian Journal

FULL PAPER

BTAIJ, 10(23), 2014 [14441-14448]

Formal verification and attack sequence generation of cryptographic protocol based on CP-nets

Bai Yunli*, Yang Ting, Mi Xiaoqin
College of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot 010018, (CHINA)
E-mail: baiyl@imau.edu.cn

ABSTRACT

An attack sequence is an execution sequence that is guaranteed to lead to a failure if the cryptographic protocol model under verification does not meet its security properties. The efficiency of generating the attack sequences for a cryptographic protocol model N is improved if N has a sub-model of security property violation events, which effectively reduced the searching scope of the state space of the protocol model. Previous work shown that, cryptographic protocol formal modelling and attack sequence generation method using CP-nets is based mainly high complexity of the whole state space search methods. This paper presents an improved security property violation events based model validation and attack sequence generation method, using the method for modelling and analyzing NS protocol, experimental results are used to evaluate the proposed method.

KEYWORDS

CP-nets; Security protocol; Attacker model; NS protocol.

INTRODUCTION

Model checking methods for the analysis of cryptographic protocols have been developed and applied widely, which not only verified the known attacks, but also discovered many new attacks. However, model-checking method faced with the exponentially increasing state space when the scale of protocol increases, the time and space they need often exceeds available resources. The attacker's capabilities include generating as much attack data as possible by the known information and the current intercepting data, resulting in protocol analysis state space increases. The main reason to generate large amounts of data is the Dolev-Yao attacker model [1], as Interrogator tool [2], NRL protocol analyzer [3].

In the current model checking method, although it is possible to generate the attack sequence, however, most formal verification methods of cryptographic protocol violating their security properties give only one attack sequence [4-10]. NRL analyzer [11] and Proverif [12] can generate multiple attack sequences based on pre-set, but they include many false attacks. Model checking methods described above are mainly off-the-fly method to generate a path to find all paths from the initial state to an insecurity state in state spaces. The complexity of generation algorithm is very high, and the unsafe state is not to be analyzed, may generate false attacks. Test case generation methods in conformance test based on the properties are used extensively. Article [13-15] proposed conformance test cases generation method for linear properties, which effectively control the test case generation scale in the finite-state, and the test cases covered the test requirements.

For formal verification of cryptographic protocols model and attack sequence generation needs, and avoiding the all-state space searching method to generate the attack sequences, this paper proposed a new formal verification and attack sequence generation method of cryptographic protocol based on CP-nets[16-18], and presented an attack sequence generation oriented CP-nets model--ASG-CPN. Using security property violation events to classify the unsafe state in the state space of protocol model, proposed an ASG-CPN based attack sequence generation method, reducing the state number of state space search and avoiding generating false attacks.

ASG-CPN MODEL

Firstly, the formal definition of CP-nets model and its related terms based on K Jensen [17,18] are given, and then address the definition of ASG-CPN. CPN Tools are specific modelling and analysis tools for CP-nets model [19].

CP-NETS model

Definition 1 (CP-nets): A *coloured Petri nets (CP-nets)* is defined as a many- tuple CP-nets= <□,, *P, T, A, V, C, G, E, I*> where:

1. □is a finite set of non-empty types, called colour sets.
2. P is a finite set of places.
3. T is a finite set of transitions with P □T = □.
4. A is a finite set of directed arcs: A □P x T □T x P.
5. V is a finite set of typed variables such that *Type*[*v*] □ Σ for all variables *v* □ *V*.
6. C is a colour function, C: P □□, p □P, C(p)$_{MS}$ □□.
7. G is a guard function, G: T □EXPR$_{V}$, that assigns a guard to each transition *t* such that Type(G(t)) = Bool.
8. E: P x T □T x P □EXPR$_{V}$, is an arc expression function, that assigns an arc expression to each arc *a* such that Type(E(a)) = C(p(a))$_{MS}$, p(a) is the place of arc a.
9. I: P□ EXPR$_{□}$, is an initialization function that assigns an initialization expression to each place *p* such that Type(I(p)) = C (p)$_{MS}$.

Definition 2 (related concepts with CP-nets):

1. A marking is a function M that maps the place p □ P into a multiset of tokens M(p)□*C*(*p*)$_{MS}$.
   The initial marking M_0 is defined as M_0(*p*) = *I*(*p*) for all *p*□*P*. The deadmarking M_d is a marking in which no transitions are enabled.
2. The variables of a transition *t* are denoted as *Var*(*t*) ⊆ *V*, consist of the free variables appearing in the guard of t or in the arc expressions of the arcs connected to *t*.
3. An observable place set P^o⊆P is the set of observable places with cardinality n_1, satisfying $0 \le n_1 \le n$ (n is the cardinality of P). An observable place *p*□P^o is the post-set of an input transition or an output transition to show what data should be observed after firing those transitions.
4. A binding of a transition t is a function b that maps the variable v□Var(t) into a value b(v)□Type[v]. The set of all bindings of a transition t is denoted by B(t). A binding element is a pair (*t,b*), *t*□*T* and *b*□*B*(*t*). The set of all binding elements *BE*(*t*) of a transition *t* is defined as *BE*(*t*)=*{*(*t,b*)|*b*□*B*(*t*)*}*.
   The set of all binding elements in a CP-nets model is denoted by *BE*. A binding elements (*t,b*) is enabled in a marking *M* when guard of the binding element (*t,b*) is true and for all the token needed by the firing of the transition t must exist in its input places, and the reached marking M' is denoted by *M[t,b>M'*.
5. A step *Y* □ *BE*$_{MS}$ is a non-empty, finite multiset of binding elements. A step *Y* □ *BE*$_{MS}$ is enabled in a marking *M* when the guard of all binding elements in the step Y are true and for all the token needed by the firing of the transition in the step Y must exist in the marking of its input places.

Definition 3 (state space): The state space of a CP-nets model with initial marking M_0, is a directed graph OG = (V, A, N) where:

V = $[M_0>$;

A = {(M1,(t,b),M2)□V×BE×V| M1[t,b>M2};

∀a□A, a=(M1,(t,b),M2) :N(a)=(M1, M2).

Here V is the set of nodes, called states (reachable markings), A is the set of edges (occurrence of binding elements), and N is a function relating edges to their end-point nodes.

Definition 4 (firing sequence): A firing sequence S *of* a CP-nets model from initial marking M_0 is a sequence of transitions $S = t_{s1}\ t_{s2} \cdots t_{sk}$, t_{si} □ T, such that $M_0[t_{s1} >M_1[t_{s2} >M_2 \cdots [t_{sk} >M$; this is denoted by $M_0[S>M$.

Definition 5 (occurrence sequence): A finite occurrence sequence of length $n \geq 0$ is an alternating sequence of markings and steps, as follows:

$$M_0 \xrightarrow{Y_1} M_1 \xrightarrow{Y_2} M_2 \cdots M_{n-1} \xrightarrow{Y_n} M_n$$

Such that $M_{i-1} \xrightarrow{Y_i} M_i$ for all $1 \leq i \leq n$. All markings in the sequence are said to be reachable from M_0. An infinite occurrence sequence is a sequence of markings and steps:

$$M_0 \xrightarrow{Y_1} M_1 \xrightarrow{Y_2} M_2 \xrightarrow{Y_3} M_3 \cdots$$

When generating attack sequences for cryptographic protocol, it should need to extract the occurrence sequence including the data exchanged by the protocol entities, thus extract the observable occurrence sequences.

ASG-CPN model

The place indicator vector □and transition label functions L are used to mark the observable places and input/output transitions.

The data, not included in the information exchanged by protocol entities, is discarded when generating the attack sequences, leaving only the actual sequence of messages exchanged between the protocol entity and the attacker. Therefore, an attack sequence generation oriented security protocol formal model is defined below.

Definition 5 **(**ASG-CPN Model): An ASG-CPN is defined as a tri-tuple ASG-CPN = *(G,□, L)*where:

ĩG =(N,M_0), is a CP-nets model N with initial marking $M_{0.}$ Model N has a set of places P with cardinality n and a set of transition T with cardinality m, where:

P=P^o□P^e, P^o is the observable place set, P^e is the internal place set, P^o□P^e=□;

T=T^o□T^i□$□^e$, T^o is the output transition set, T^i is the input transition set, $□^e$ is the internal transition set, T^o□T^i□$□^e$=□;

2̃□□*(□$_{1,}$□$_{2,}$...,□$_n$)^T* is the indicator vector for the place set P = {p_1, p_2,..., p_n}, with cardinality n, The indicators related to observable places are equals to 1, otherwise indicators for internal places are equals to 0:

$$\psi_i = \begin{cases} 1, \forall\ p_i \in P^o,\ 0 \leq i \leq n_1. \\ 0, \forall\ p_i \in P^e,\ 0 \leq i \leq n - n_1. \end{cases} \qquad (1)$$

3̃.L is the labelling function for transitions, L:T→□ ∪ {ε}, □is the label set(character set), used to describe the input/output information sets of the input/output transitions, for the internal transitions the label sets are empty ε:

$$L(t) = \begin{cases} e,\ \forall t \in (T^o \cup T^i),\ e \in \Lambda. \\ \varepsilon,\ \forall t \in T^e. \end{cases} \qquad (2)$$

Cryptographic protocols are concerned with exchanging messages between entities via an untrusted medium. The protocols aim at providing security properties such as confidentiality of transmitted data, user authentication etc. A cryptographic protocol is usually described as a sequence of messages, and encryption is used to achieve security goals. As an example consider the simple Needham-Schroeder (NS) protocol [20-22]to build the ASG-CPN model.

NS protocol is an asymmetric key authentication protocol, the data exchange sequence described as follows:

1. A □ B: {N_a, A}puk_B
2. B □ A: {N_a, N_b}puk_A
3. A □ B: {N_b}puk_B

Where entity A is the session initiator, B is the session responder. A □ B: {Na, A}puk_B indicates that entity A sends message of {N_a, A} puk_B to entity B. {N_a, A} puk_B means the message {N_a, A} is encrypted by puk_B, the public key of entity B, only the private key of entity B can decrypt the encrypted message. N_a, N_b is the secure nonce chose by entity A and entity B respectively.

In Figure 1, channel places c1, c2, c3 are observable places, the others are internal places; output transitions send1, send2, and send3 are used to describe the transmit events; input transitions, receive1, receive2, receive3, indicate the entity behaviours after receiving the datagram; the other transitions are internal transitions. Specific case of output datagram is obtained by tokens of observable places, which is associated with output transitions. The place indicator vectors and transition label function describe the actual features, as follows:

$$\psi_i = \begin{cases} 1, \forall\ p_i \in P^o,\ P^o=\{c1,\ c2,\ c3\},\ 1\leq i\leq 18. \\ 0, \forall\ p_i \in P^e,\ P^e = P - P^o \end{cases} \qquad (3)$$

$$L(t) = \begin{cases} BE(t), \forall t \in \{\mathit{send1, send2, send3, receive1,} \\ \qquad \mathit{receive2, receive3}\}. \\ \varepsilon, \forall t \in T^e. \end{cases} \qquad (4)$$

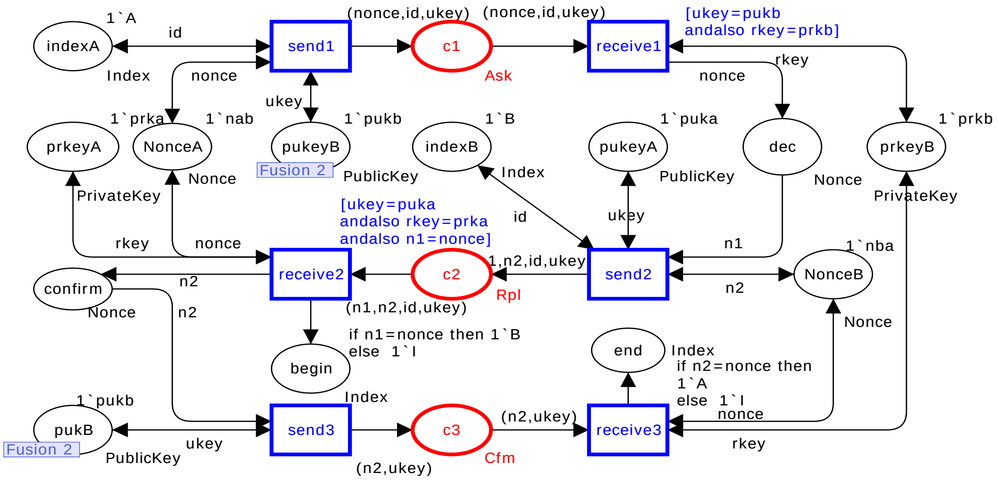
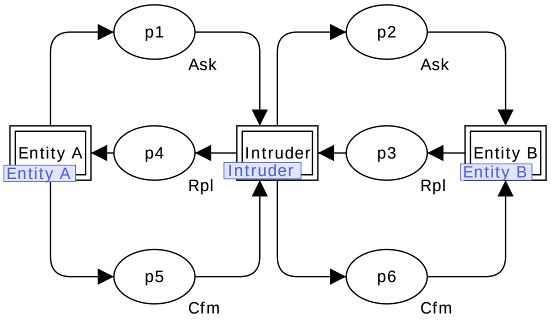


Figure 1: ASG-CPN model of NS Protocol

Here an attacker model added to the ASG-CPN model of NS protocol as illustrated in Figure 2. The observable places c1, c2, c3 are divided into two port places respectively, and add the Dolev-Yao attacker model into the two new places.

The channel place c1 is corresponding to the two port places p1 and p2, and the substitution transition int_Ask was added between the place p1 and place p2. The channel place c2 is corresponding to the two port places p3 and p4, and the substitution transition int_Rpl was added between the place p3 and place p4. The channel place c3 is corresponding to the two port places p5 and p6, and the substitution transition int_Cfm was added between the place p5 and place p6, as illustrated in Figure 2 and Figure 3。



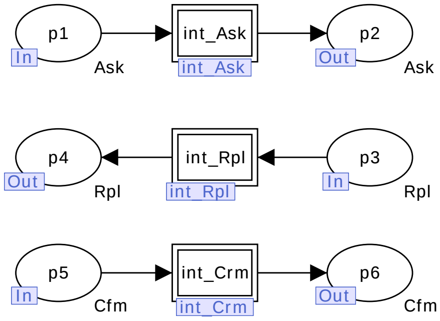


Figure 2: ASG-CPN Top-level Model of NS an Attacker

Figure 3: ASG-CPN Attacker Subpage of NS protocol with protocol

The subpage related to the substitution transition int_Ask is chose as an example to illustrate the attacker subpage model. In the Figure 4, the corresponding transitions of port places p1 and p2 are output transition t0 and input transition t4.

ATTACK SEQUENCE GENERATION METHOD

According to the cryptographic protocol specifications and features of the models, gives formal definition of security property. Using security property violation events to classifying describe the unsafe states, proposed an ASG-CPN based attack sequence generation method, reducing the searching scope of the state space, and avoiding generating false attacks.

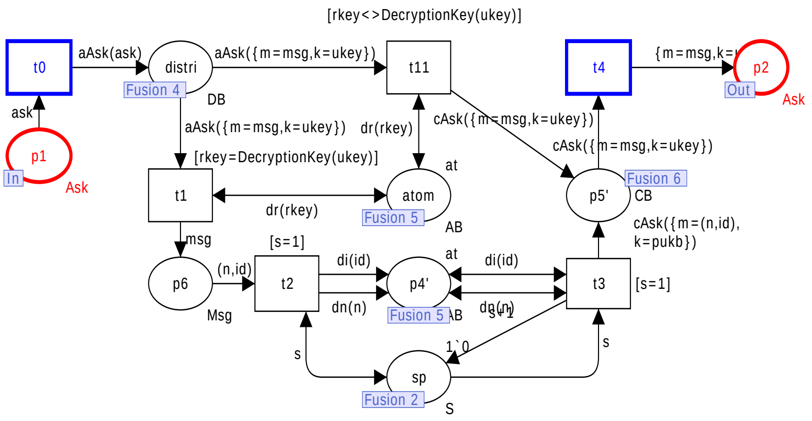


Figure 4: ASG-CPN Model of Attacker int_Ask Subpage

Security properties

The purpose of cryptographic protocol is to prevent malicious side attack from network environment to achieve the intended security goals. Those security goals are called security properties. In this section, the most common security properties, including confidentiality property and authentication property, are given their formal definition.

Definition 6 (confidentiality property): Let *N* be the formal model of the cryptographic protocol, *In* be the attacker, D_{In} be the message set places of the attacker, and m' be the confidentiality information. If there is an occurrence sequence □in the set P of occurrence sequences, let m' □D_{In}, then the protocol does not meet the confidentiality properties:

$$\exists\sigma \in P\text{ , } P=\{M_0 b_1 M_1 b_2 ...| M_0 = M \wedge M_0 \xrightarrow{b1} M_1 \xrightarrow{b2} M_2 ...\}\ \exists M_i \in \sigma, 0 \le i \le |\sigma|, \alpha(M_i)\text{=true } \textit{iff}\ m' \in M_i(DIn),\ M_i |=_{St} \tilde{\alpha}$$

In analyzing and verifying confidentiality properties, because the attacker would have been store the messages after received, therefore only need to consider the deadmarking states of a finite-length occurrence sequence, whether the attacker access to confidential information. The state space (reachability graph) of the cryptographic protocol CP-nets model, does not meet the confidentiality properties can be described as:

$$\exists\sigma \in P\text{ , } P=\{M_0 b_1 M_1 b_2 ...Md| M_0 = M \wedge M_0 \xrightarrow{b1} M_1 ... \xrightarrow{bn} Md\}\text{ , } Md \in \sigma, \alpha(Md)\text{=true } \textit{iff}\ m' \in Md(DIn),\ Md|=_{St}\alpha\ .$$

Among them, M_d is the terminating state of each occurrence sequence. If M_d contains confidential information, it indicates that the cryptographic protocol dose not meet its confidentiality property.

Definition 7 (authentication property): Given an authentication cryptographic protocol model N includes session initiator A, responder B and an attacker In, begin (M`) event indicates the event when entity A authenticate the entity B, end(M``) indicates the event when entity B authenticate the entity A, in the occurrence sequences of model N, when entity B performs end (M``) event, entity A has also been already performed begin (M`), called the cryptographic protocol model N satisfies authentication property.

For the NS authenticated key exchange protocol, using begin(M`) event and end(M``) event represents the events entities A and B authenticates each other, the variable is set to reach a mutual authentication requires the need for both sides to reach a shared key $Share_A$ and $Share_B$, the NS protocol satisfy the authentication property, when the following conditions are met：

(1) $Share_A = Share_B$;

(2) begin(M`)□□$_I$,end(M``)□□$_k$,$M_0 \overset{\sigma}{\Rightarrow}$ □$_k$,□$_I$□$\tilde{□}$

Attack sequence generation

After the modelling cryptographic protocol system using CP-nets, it can use ASKCTL[23] to verify the model, and the system properties are able to give “satisfy” or “not satisfy” conclusions. However, if the verification result is "not satisfy", ASKCTL validation process does not give further diagnostic information, so that users cannot accurately find the places where the error occurred, and cannot discovery which transition triggering sequence can lead to property does not meet.

This section presents attack sequence generation method of the cryptographic protocol model, which can effectively gives the occurrence sequence of protocol model when the security properties are not meet, and can provide sufficient information for protocol designers and analysts, to propose protocol improvement scheme.

Currently, the most common attack sequence generation methods are mainly based on whole state space search method. Due to the state space explosion problem, most studies have focused on the single session or multiple sequence sessions scenarios. This article focused on multiple concurrent sessions cryptographic protocol model validation and attack sequence generation problem, Using security property violation events to describe the non secure states in the state space of the cryptographic protocol formal model, proposed an attack sequence generation method based on ASG-CPN model, which is reducing the search range of state space, and avoiding false attacks。

Attack sequence generation method can also be based on marking the insecurity states, which can effectively reduce the search range of the state space. The insecurity states can be classified using security property violation events, and the false attacks state can be ruled out.

Firstly, the security property violation events are defined, and then the attack sequence generation method based on ASG-CPN model including its practical applications is addressed.

Definition 8 (Security Property Violation Events, SPVE): Security property violation events describe the behaviour sequences of cryptographic protocol contrary to the security properties. State PR of SPVE is the Initial state on behalf of the successful implementation of the events, State PE of SPVE is the terminating state of the events. The initial states and terminating states are described by the marking of protocol model.

The SPVE are described as a sub-model of the cryptographic protocol system model, such that place set of the SPVE is a subset of the protocol model. The each place in the SPVE can correspond with the place in the system model, and the place components of each marking in the SPVE can correspond with the place components of the states in the state space of the system model.

Definition 9 (State projection): Given a CP-nets model N, with initial marking M_0, and state space OG = (V,A,N), V is the node set, called reachable markings. For each place subset $P_S \subseteq P$, $m1 \subseteq M(P_S)$, p□P_S, $M_i(p)$□V, m1(p)□ $M_i(p)$, then the state projection of ml onto the reachable markings in the N, denoted as PROJ(m1↓V) =$M_i(p)$.

Perform state projection operation to initial and terminating state of SPVE respectively, denoted as si=PROJ(PI↓V), Se=PROJ(PE↓V).

Definition 10 (Execution Sequences of SPVE): For a SPVE of the cryptographic protocol, with initial state PR and terminating state PE, the execution sequences of SPVE, denoted as $T_{sub}(M_0,PR,PE)$, in which $M_0[T_{sub'}>PR$ and $M_0[T_{sub}>PE$, $T_{sub'} \subseteq T_{sub}$.

For a security protocol model, firstly use model validation techniques to find that the model does not meet the security properties, and then search the initial state and final state of the SPVE in the state space, given the sub-path that contains all the sequences between the two states. Because only need to consider the data exchanged between the entities, so generating attack sequences retain the information of observable places, and the information of the internal places can be discarded.

Definition 11(Observable occurrence sequence set): Given an ASG-CPN =*(G,*□,L) model and the $T_{sub}(M_0,PR,PE)$ of a SPVE , T_{obs}=Obs(T_{sub}) extract the observable execution sequence T_{obs} from T_{sub}, T_{obs} is an firing sequence set corresponding with each sequence in the $T_{sub,}$ retaining the binding elements of the input/output transitions, and each state retaining the elements of the observable places.

Table 1 shows the attack sequence generation algorithm based on ASG-CPN Model. Firstly, it performs state projection pr and pe of SPVE onto reachability graph OG of the ASG-CPN model, obtained PROJ(pr↓V) and PROJ(pe↓V); if PROJ(pr↓V)=Φ or PROJ(pe↓V)= Φ then quit, else denoted as si, se respectively; and then searches the execution sequences $T_{sub}(M_0,Si,Se)$ from the initial marking according to the indicator vector □and label function L of ASG-CPN model, and obtain the corresponding observable sequences Obs(tr(M_0,Si,Se)).

In the algorithm, $T_{sub}(M_0,Si,Se)$ and Obs(tr(M_0,Si, Se))involved state space searching problem, the specific search process is as follows:

- $T_{sub}(M_0,Si,Se)$ use Breadth-first search algorithm to search the all firing sequences initiated from initial state M_0 , via state Si, and reach terminating state Se in the state space;
- Obs(tr(M0,Si,Se)) obtains the each observable occurrence sequence tr(M0,Si,Se) ⊆Tsub(M0, Si,Se), according to the indicator vector □and label function L of ASG-CPN model, and gets the attack sequences.

TABLE 1: Attack Sequence Generation Algorithm based on ASG-CPN Model

```
Input:  initial state Pr and terminating state Pe of SPVE,
        cryptographic protocol model ASG-CPN=(G,□,L), OG = (V,
        A, N)
Output: Attacksequences
Begin
    Si=PROJ(pr↓V);
    Se=PROJ(pe↓V);
    If Si!=Φ and Se!=Φ then
        If $T_{sub}(M_0,$  Si,  Se)!=Φ then
            For □tr□$T_{sub}(M_0,Si,Se)$
            Attacksequence= Attacksequence^^ Obs(tr($M_0$,Si,Se)); /
Next
    End
    End
    Output(Attacksequence);
End
```

Algorithm Analysis: By all occurrence sequences between to the initial and final states in the state space to generate attack sequences, its complexity is O(n*m), the number of terminal state is m, the total number of state is n. The attack sequence generation method based on SPVE has effectively controlled the search number of the state space; meanwhile avoided generating false attacks.

For the NS authenticated key exchange protocol, using begin(M`) event and end(M``) event represents the events entities A and B authenticates each other, the variable is set to reach a mutual authentication requires the need for both sides to reach a shared key $Share_A$ and $Share_B$, the NS protocol satisfy the authentication property, when the following conditions are met :

(1) $Share_A = Share_B$;

(2) begin(M`)$\square\square_I$,end(M``)$\square\square_k$,$M_0 \overset{\sigma}{\Rightarrow} \square_k,\square_I\square\tilde{\square}$

Below using 1995 lowe [22] released NS protocols middle attack, as an example to demonstrate the attack trace generation method based on SPVE. Based on ASG-CPN models and SPVE model of the NS protocol (Figure 5), indicate the attack sequence generating process.

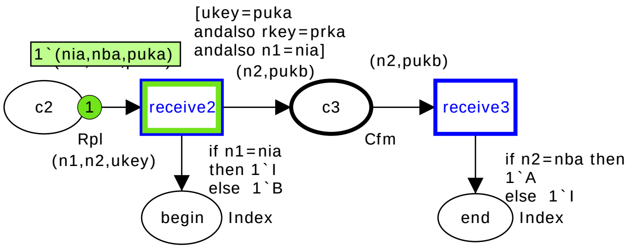


Figure 5: SPVE Model of Man-in-Middle Attack on NS Protocol

A SPVE described in Figure 5 is that an entity B sent authentication acknowledgment data to entity A, and then entity A need to return authentication confirmation data; Place “begin” stores identity I for authenticated session responder of entity A; place “end” stores identity A for authenticated session initiator of entity B.

With the state space of NS protocol ASG-CPN model generated by CPN Tools, after performing state projection of initial state and final state of the SPVE in Figure 5, PR={63}, PE={137,138}. Based on the state reachability, calculates all possible execution sequences: $T_{sub}(M_0,$ PR,PE).

By searching T_{sub}(PR,PE), the execution sequences include total 66 sequences, such as $M_{63}TM_{68}$ $TM_{73}TM_{78}$ $TM_{83}TM_{91}TM_{101}TM_{114}TM_{125}TM_{133}TM_{13,}$ in which the transition names are omitted. But $T_{sub}(M_0, M_{63})$. The execution sequences from initial marking $M_0$to the initial state PR of SPVE, has only one sequence, $M_1TM_2TM_3$ $TM_5TM_7TM_9TM_{13}TM_{17}TM_{25}TM_{34}TM_{42}TM_{47}TM_{52}TM_{57}$.Therefore, there exist 66 execution sequences in $T_{sub}(M_0,$ PR,PE). According to the observable place indicator and input/output transition label, the one man-in-middle attack sequence of NS protocol was obtained [22].

CONCLUSION

In this paper, formal verification of cryptographic protocols and attack sequence generation method conducted in-depth research. The ASG-CPN model for the attack sequence generation was proposed, which added input-output transitions and observable places in the original cryptographic protocols CP-nets model. For a comprehensive analysis of cryptographic protocol attack, the SPVE is given based on in-depth analysis of security properties. An attack sequence generation method based on SPVE was proposed. Through the formal verification and attack sequence generation practice of NS protocol, fully illustrates the effectiveness of new and improved methods.

ACKNOWLEDGEMENT

The Program of Natural Science Foundation of Inner Mongolia of China under Grant No. 2012MS0914 supported this work.

REFERENCES

[1] Dolev D Y A. On the security of public key protocols. Proceedings of the IEEE 22nd Annual Symposium on Foundations of Computer Science, Nashville, TN. 1981. p. 350-357.

[2] Millen J K, Clark S C, Freedman S B. The interrogator: Protocol security analysis. Software Engineering, IEEE Transactions on, 1987(2): p. 274-288.

[3] Meadows C. The NRL protocol analyzer: An overview. The Journal of Logic Programming, 1996. 26(2): p. 113-131.

[4] Lowe G. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. Tools and Algorithms for the Construction and Analysis of Systems, 1996: p. 147-166.

[5] Vigano L. Automated security protocol analysis with the AVISPA tool. Electronic Notes in Theoretical Computer Science, 2006. 155: p. 61-86.

[6] Maggi P, Sisto R. Using SPIN to verify security properties of cryptographic protocols. Model Checking Software, 2002: p. 85-87.

[7] Lowe G, Roscoe B. Using CSP to detect errors in the TMN protocol. Software Engineering, IEEE Transactions on, 1997. 23(10): p. 659-669.

[8] Shmatikov V, Mitchell J C. Finite-state analysis of two contract signing protocols. Theoretical Computer Science, 2002. 283(2): p. 419-450.

[9] Al-Azzoni I, Down D, Khedri R. Modelling and verification of cryptographic protocols using coloured petri nets and design/CPN. Nordic Journal of Computing, 2005. 12(3): p. 201-208.

[10] Mitchell J C, Mitchell M, Stern U. Automated analysis of cryptographic protocols using Murφ. Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on. 1997: IEEE. p. 141-151.

[11] Meadows C. The NRL protocol analyzer: An overview. The Journal of Logic Programming, 1996. 26(2): p. 113-131.

[12] Blanchet B. An efficient cryptographic protocol verifier based on Prolog rules. Proceedings of the 14th IEEE workshop on Computer Security Foundations. 2001. p. 82.

[13] Fernandez J-C, Mounier L, Pachon C. Property oriented test case generation. Formal Approaches to Software Testing, 2004: p. 1101-1102.

[14] Sun T, Ye X, Liu J. A Test Generation Method Based on Model Reduction for Parallel Software. Proceedings of the 2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies. 2012: IEEE Computer Society. p. 777-782.

[15] SUN T, YE X. A Model Reduction Method for Parallel Software Testing. Journal of Applied Mathematics, 2013.

[16] Kristensen L, Petrucci L. An approach to distributed state space exploration for coloured Petri nets. Applications and Theory of Petri Nets 2004, 2004: p. 474-483.

[17] Jensen, K., L. Kristensen, and L. Wells, Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. International Journal on Software Tools for Technology Transfer (STTT), 2007. 9(3): p. 213-254.

[18] Jensen K, Kristensen L. Coloured Petri Nets: Modelling and Validation of Concurrent Systems. 2009: Springer-Verlag New York Inc.

[19] CPN Tools[EB/OL]:http://cpntools.org .

[20] Needham R M, Schroeder M D. Using encryption for authentication in large networks of computers. Communications of the ACM, 1978. 21(12): p. 993-999.

[21] Needham R M, Schroeder M D. Using encryption for authentication in large networks of computers. Communications of the ACM, 1978. 21(12): p. 993-999.

[22] Lowe G. An attack on the Needham-Schroeder public-key authentication protocol. Information processing letters[J], 1995. 56(3): p. 131-133.

[23] Cheng A, Christensen S, Mortensen K H. Model checking coloured petri nets exploiting strongly connected components. DAIMI PB, 1997. 26(519): p. 1-17.