2010

Security Analysis of Session Initiation Protocol - A Methodology Based on Coloured Petri Nets

Lin Liu
University of South Australia

SECURITY ANALYSIS OF SESSION INITIATION PROTOCOL - A METHODOLOGY BASED ON COLOURED PETRI NETS

Lin Liu

School of Computer and Information Science
University of South Australia
Mawson Lakes, South Australia
lin.liu@unisa.edu.au

Abstract

In recent years Voice over Internet Protocol (VoIP) has become a popular multimedia application over the Internet. At the same time critical security issues in VoIP have started to emerge. The Session Initiation Protocol (SIP) is a predominant signalling protocol for VoIP. It is used to establish, maintain and terminate VoIP calls, playing a crucial role in VoIP. This paper is aimed at developing a Coloured Petri Net (CPN)-based approach to analysing security vulnerabilities in SIP, with the ultimate goal of achieving a formal and comprehensive security assessment of SIP specification, and creating a platform for evaluating countermeasures for securing SIP. In the paper we present a method for modelling the behaviour of SIP and its security threats using CPNs, and discuss suitable techniques for analysing the CPNs for investigating SIP security issues. The CPN models and the analysis techniques will then become the platform for analysing the behavior of SIP that is enhanced with proposed security countermeasures.

Keywords: Voice over IP, Session Initiation Protocol, security analysis, Coloured Petri Nets, protocol verification

INTRODUCTION

Voice over Internet Protocol (VoIP) has been rapidly deployed in recent years, due to its lower cost and greater flexibility comparing to Public Switched Telephone Networks. Before a VoIP call can take place, signalling protocols must be employed to establish a session, and to maintain and terminate the established session. Currently a dominant VoIP signalling protocol is the Session Initiation Protocol (SIP) developed by the Internet Engineering Task Force, and the specification of SIP is published in RFC 3261 (Rosenberg et al 2002). Besides its increasing popularity in VoIP, SIP has been adopted by the 3rd Generation Partnership Project as a signalling protocol and permanent element of the IP Multimedia Subsystem architecture (Sparks 2007).

Multimedia information tends to attract more attentions and raise curiosities from intruders as people are keen on viewing and hearing communications. More importantly, VoIP protocols, including SIP, were designed without serious security concerns in mind, and the open architecture of the Internet makes attacks easier. Consequently, along with the widespread deployment of VoIP, its security flaws have emerged and become a problem being addressed, in the deployment of VoIP systems, and in the research and development of VoIP techniques (Dantu et al 2009, VOIPSA 2010).

The security issues of SIP have been investigated a great deal recently (Geneiatakis et al 2006, Sisalem et al 2009, Werapun et al 2009, Zhang 2007). Many intrusion detection methods and security countermeasures have been proposed (Geneiatakis et al 2006, Sisalem et al. 2009, Werapun et al 2009, Zhang 2007, Ehlert et al 2010, Ormazabal et al 2008, Abdelnur et al 2009, Sengar et al 2006, Ding and Su 2007). Most articles discuss or identify SIP security holes through critical reviews or experiments of typical SIP application scenarios. Evaluations of proposed detection methods and countermeasures largely rely on experimenting with real VoIP networks or test beds. Some of the intrusion detection systems make use of formal methods, such as communication state machines (Sengar et al 2006) and Petri nets (Ding and Su 2007).

However, to our best knowledge, little work has been done on using formal methods to systematically and comprehensively analyse security vulnerabilities of the SIP specification in RFC 3261. Although SIP has been widely deployed, the study of SIP security is still immature. So it is important to not only look into security problems and solutions for SIP-based networks, but also to conduct formal security analyses of SIP design. More importantly security analyses based on experiments only are not comprehensive, as we cannot test each and every possible scenario, different operator networks may have different settings, and the implementations of

SIP may not be the same. On the other hand, formal methods, such as Coloured Petri Nets (CPNs) (Jensen 1997, Jensen et al 2007), allow us to obtain rigorous and more complete analysis results.

CPNs have been applied widely in verifying communication protocols, business processes, and some other systems (The CPN Tools 2010, Billington et al 2004). Related to the work in this paper, are the applications of CPNs (and other forms of Petri nets) to verifying security protocols, particularly, cryptographic protocols (Nieh and Tavares 1993, AI-Azzoni et al 2005, Ding and Su 2008, Permpoontanalarp and Sornkhom 2009).

In this paper, we apply and further develop the basic idea in (Nieh and Tavares 1993) for verifying cryptographic protocols, for security analysis of SIP. The methodology formed in this paper, and our future work, are aimed at a comprehensive and formal security analysis of SIP design, to provide theoretical support to known security threats, and to discover new security holes of SIP. Another goal of our work is to use the CPN models for SIP and the intruders as a platform to evaluate security countermeasures of SIP.

The rest of the paper is organised as follows. Section 2 is an overview of SIP and the identified SIP security threats; Our CPN-based methodology for SIP security analysis is described in Section 3. Section 4 concludes the paper and discusses future research plan.

SIP AND ITS SECURITY RISKS

SIP

SIP is a signalling protocol for establishing, modifying and terminating a multimedia session between two or more participants. These services are provided by SIP components (entities), including user agent, proxy server, redirect server, and registration server.

SIP has a layered architecture, comprising the syntax and encoding, transport, transaction, and transaction user (TU) layers (Fig. 1).

The syntax and encoding layer specifies the format and structure of SIP messages. A SIP message can be a request from a client to a server, or a response from a server to a client. For each request, a method (such as INVITE or ACK) must be carried to invoke a particular operation on a server. For each response, a status code is declared to indicate the acceptance, rejection or redirection of a request (Table 1).

The second layer of SIP is the transport layer. It defines the behaviour of SIP entities in sending and receiving messages over the network.

Two types of SIP transactions are defined for the transaction layer, the INVITE and the non-INVITE transactions. An INVITE transaction is initiated when an INVITE request is sent; and a non-INVITE transaction is initiated when a request other than INVITE or ACK is sent. Each of the transactions consists of a client transaction sending requests and a server transaction responding to requests.

The top layer holds the Transaction Users (TUs), which can be any SIP entity except a stateless proxy.

SIP is a transaction-oriented protocol that carries out tasks through different transactions. So among the four SIP layers, the transaction layer is the most important one. It is responsible for request-response matching, retransmission handling with unreliable transport medium, and timeout handling. To accomplish a transaction, the transaction (such as the INVITE transaction) in the transaction layer is required to interact with the TU and the SIP transport layer.

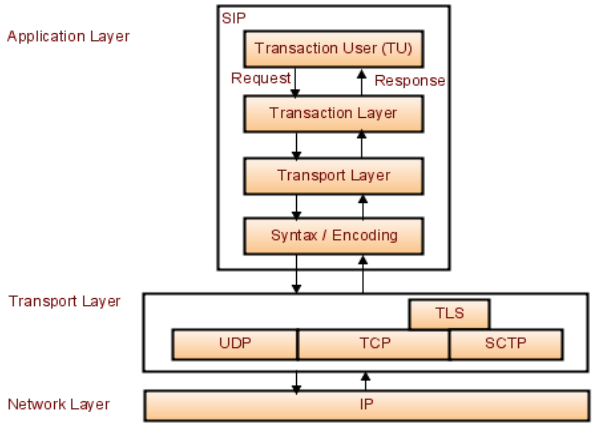

Fig. 1. Layered structure of SIP (Ding and Liu 2008)

Table 1. SIP response messages (Rosenberg, J., et al 2002)

Response	**Functions**
1xx (100-199)	Provisional - indicate the request was received but not yet accepted.
2xx	Success - indicate the request was received successfully and accepted
3xx	Redirection - a further action is required to complete the request
4xx	Client Error - bad syntax found in the request which cannot execute at this server
5xx	Server Error - the server failed to answer the request
6xx	Global Failure - no server can answer the request

Fig. 2 is an example showing how the INVITE and non-INVITE transactions are applied in a SIP call. Initially, the client makes a call to the server by sending an INVITE request (without a proxy server being involved). The server can decide to accept, redirect or reject the INVITE from the client. In this example, the server does not answer the call immediately. A 180 ringing response is sent to the client. Eventually, the server accepts the INVITE by sending a 200 OK response to the client. Once the client receives this response, an acknowledgement is sent to the server. The session is established. Then they start the audio/video talk. The audio/video data are transmitted by RTP (Real Time Transport Protocol). When the client hangs up the call, a BYE message is sent to the server. The server confirms the receipt of the BYE message with a 200 OK response, which terminates the call.

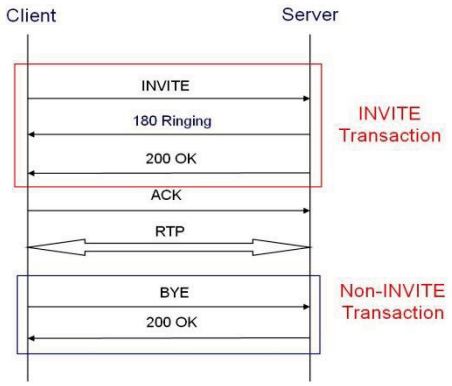

Fig. 2 An example SIP call flow

Security risks in SIP

As mentioned previously SIP was designed without having security as a primary concern. In SIP specification (Rosenberg et al 2002), no new security mechanisms are defined. Instead, SIP recommends the adoption of existing security models used over the Internet (Rosenberg et al 2002). It has been found that even with certain security mechanisms, SIP are still facing severe security challenges, leaving VoIP networks to security risks (Dantu et al 2009, Geneiatakis et al 2006).

In SIP specification (Rosenberg et al 2002), a number of common security threats are listed, including: registration hijacking, impersonating a server, tampering with message bodies, tearing down sessions, and denial of service and amplification. Unfortunately the descriptions are very general, without sufficient details and analyses of the threats. Therefore, great efforts have been made to elaborate on those threats and to create solutions to them (Geneiatakis et al 2006, Sisalem et al 2009, Werapun et al 2009, Zhang 2007, Ehlert et al 2010, Ormazabal et al 2008, Abdelnur et al 2009, Sengar et al 2006). We would like to add to such efforts by looking into SIP security issues with the help of formal methods.

THE CPN-BASED METHODOLOGY FOR SIP SECURITY ANALYSIS

In this section we introduce our approach to security analysis of SIP. It is based on the modelling constructs and the state space analysis technique of Coloured Petri Nets (CPNs) (Jensen 1997, Jensen et al 2007).

Coloured Petri Nets

Petri Nets (PNs) (Murata 1989) are a formal technique suitable for modelling and analysing distributed systems that are characterised by concurrency and nondeterminism. A major strength of PNs is their support for the analysis of many properties of the systems being modelled. The graphical representation of a PN and its executable nature make it an attractive and easy to understand formal technique.

Various forms of PNs have been devised to investigate systems with different levels of complexity. The CPNs introduced by Jensen (Jensen 1997, Jensen et al 2007) enhance basic PNs with the strength of a high-level programming language, allowing multiple data types to be defined and the values of these data types to be manipulated, and larger and more complex systems to be modelled. The hierarchical representations of CPNs enable the creation of modularised models. We choose to use CPNs due to their many successful applications in protocol verification (Jensen et al 2007, The CPN Tools 2010, Billington et al 2004), and their well-developed supporting software package, the CPN Tools (Jensen et al 2007).

The graphical representation of a CPN consists of sets of nodes of two types, *places* (ellipses in Fig. 3) and *transitions* (rectangles), with *arcs* connecting places to transitions and transitions to places.

A place can have a type, called a *colour set.* For example, in Fig. 3, place **Client** has the colour set **STATEC**. The colours chosen from its colour set and associated with a place are called *tokens.* The multiset of token(s) on a place is a *marking* of this place and the distribution of tokens on all the places of a CPN is a marking or *state* of the CPN. The initial tokens are known *as initial markings* of these places (shown under or above a place). The initial marking of the CPN in Fig. 3 is (**Client**: 1`calling; **Server**: 1`idle; **Request**s: 1`[]; **Responses**: 1`[]; **INVITE Sent**: 1`0), indicating that initially the INVITE Client transaction is calling, the Server transaction is idle, the communication channels in both directions are empty (respresented by an empty list []), and the number of INVITE requests that have been sent is zero. Here, 1` shows that a value appears once in a multiset (Jensen et al 2007).

A transition can be *enabled* if each of its input places (place having an arc attached from it to this transition) has sufficient tokens according to the *arc inscriptions* (conditions inscribed on respective arcs) and if the transition *guard* evaluates to true. A guard is given in square brackets next to a transition. In Fig. 3, the only enabled transition in the initial marking is **Send Request.** When a transition is enabled, it may *occur,* and it then removes tokens from its input places and creates new token in its output places according to arc inscriptions, and the model moves into a new state. *Variables* can be used in arc inscriptions and the guard. They are bound to values upon the occurrence of the given transition.

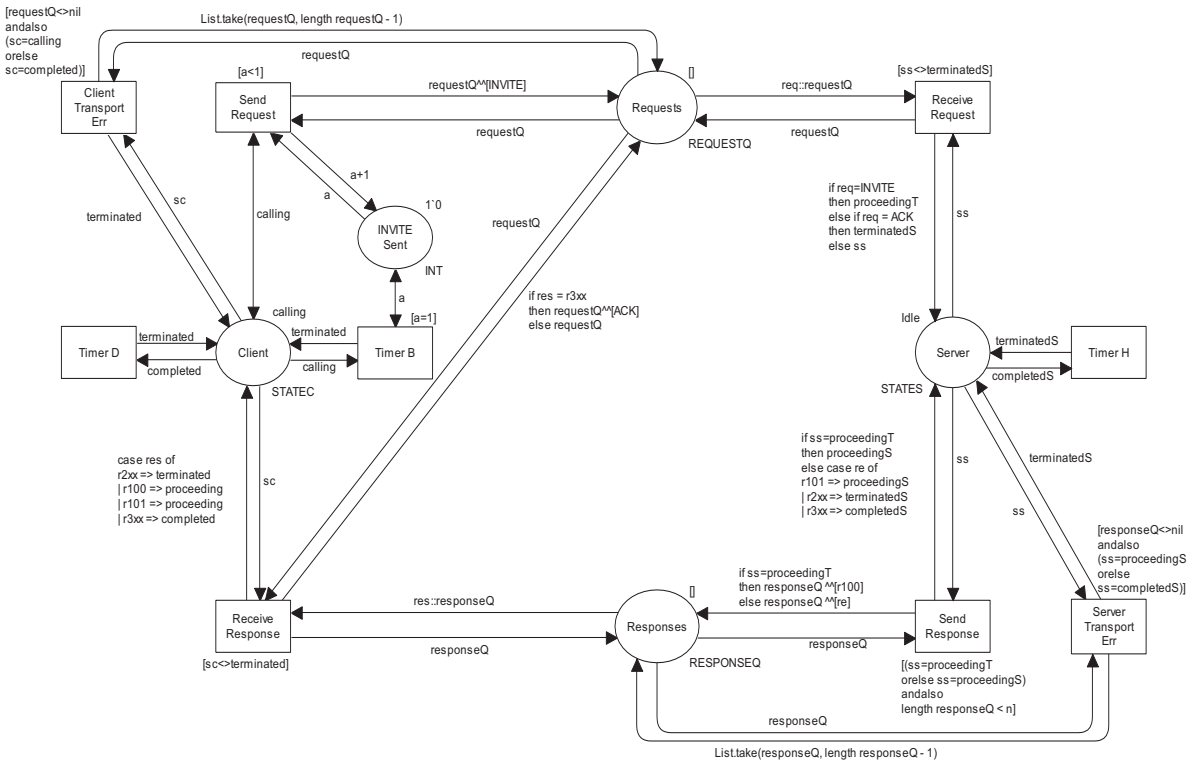

Fig. 3 A CPN model of the SIP INVITE transaction (Ding and Liu 2008)

With large CPNs or models with parts that could be reused, we can represent the parts (subnets) using *substitution transitions,* which interact with other parts of the CPN through a set of input/output *socket* places. Substitution transitions can be reused, i.e. one substitution transition can have multiple instances**,** and the marking of each instance can be completely independent from the markings of other instances of the same substitution transition. We will make use this feature of CPNs in our modelling.

State space analysis is a main analysis technique for CPNs (Jensen 1997, Jensen et al 2007). The state space of a CPN is a directed graph comprising all reachable markings (states) and state changes of the CPN model. By generating and querying the state space using supporting tools, we can verify the properties of a modelled system, such as absence of deadlocks (undesirable terminal states), whether or not a given state can be reached or a required service can be delivered.

Modelling SIP and its security threats

1) Overall model

In (Nieh and Tavares 1993) the authors proposed an approach to verifying cryptographic protocols using a coloured Petri net which is less expressive than the CPN introduced by Jensen (Jensen 1997). This verification approach was used in (AI-Azzoni et al 2005, Ding and Su 2008, Permpoontanalarp and Sornkhom 2009) with Jensen's CPN, for verifying security protocols. In this paper, we extend the approach from the domain of security protocol verification to the verification of SIP (which is not a security protocol) with the presence of security threats.

The general modelling idea of (Nieh and Tavares 1993) is illustrated in Fig. 4. With this approach, instead of modelling various scenarios of security attacks explicitly, two types of objects, protocol entities and intruders, are identified. By modelling the behavior of protocol entities and an intruder, security threats are captured in the model implicitly.

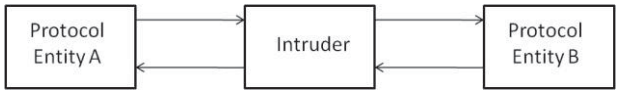

Fig. 4 Protocol entities with an intruder – a general model (Nieh and Tavares 1993)

Utilising this idea, we create the top-level CPN model for SIP security analysis (Fig. 5). In this model, the two substitution transitions on the left and right hand sides (**Client** and **Server**) are for the operations of SIP at the client and server sides respectively. The substitution transition in the middle (**Intruder**) captures the behavior of an intruder. The two places (**CS_Channel** and **SC_Channel**) model the communication channel from client to server and the channel in the reverse direction respectively.

In (Nieh and Tavares 1993) and (AI-Azzoni et al 2005, Ding and Su 2008 , Permpoontanalarp and Sornkhom 2009), corresponding to one message channel, the intruder module (substitution transition) has one pair of input and output socket places (i.e. two separate socket places). With our model, for one channel, the intruder substitution transition's input and output socket places are the same. This halves the number of places for modelling channels, which could significantly reduce the sizes of the state spaces. As state space explosion is the "killer" problem with state space analysis (Jensen et al 2007), it is important to have CPN models with smaller numbers of places.

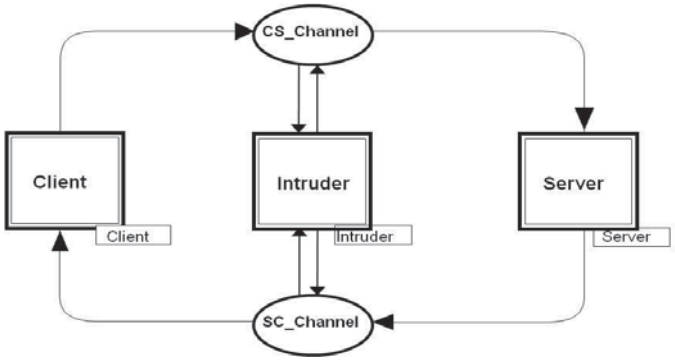

Fig. 5 The top-level CPN model of SIP with an intruder

2) CPN model for SIP

As mentioned before, SIP carries out tasks through different transactions, including INVITE and non-INVITE transactions. Correspondingly in our CPN model, the top-level **Client** substitution transition (Fig. 5) will include two second-level substitution transitions: **Client_INVITE**, and **Client_nonINVITE**; and the top-level **Server** substitution transition will have two second-level substitution transitions: **Server_INVITE**, and **Server_nonINIVITE**.

In (Ding and Liu 2008), we created a CPN model for the INVITE transaction (reproduced in Fig. 3). In the model, we use one place (**Client**) to model the INVITE client transaction and one place (**Server**) for the INVITE server transaction. Events, including sending and receiving messages, timeout and error reporting are modelled with transitions. The two places in the middle of the CPN, **Requests** and **Responses** model the channel from the client to the server and the channel in the reverse direction respectively. Relating this CPN to the top-level model in Fig. 5, its left hand side part will be mapped to the **Client_INVITE** substitution transition within the top-level substitution transition, **Client** in Fig. 5, and its right hand side part to the **Client_nonINVITE** substitution transition within the top-level substitution transition, **Server**. Moreover, places **Requests** and **Responses** are places **CS_Channel** and **SC_Channel** in our top-level model respectively.

We will use the same modelling approach to create the CPN model for SIP non-INVITE transaction, and include it as part of the top-level model, in the same way as that for the INVITE transaction model.

3) CPN model for the intruder

We assume a Dolev-Yao intruder (Dolev and Yao 1983), who has complete control of communication channels, i.e. who can intercept, drop, replay, and forge messages. For each of the actions, we model it with a substitution transition inside the **Intruder** substitution transition in Fig. 5. That is, the top-level **Intruder** substitution transition will consist of four second-level substitution transitions: **Intruder_Intercept**, **Intruder_Drop**, **Intruder_Replay**, and **Intruder_Forge**.

Since the intruder has complete control of the channels, it is proper to use the two channel places as the socket places for the **Intruder** substitution transition, so that messages can be obtained from and put into the channels by the intruder in our model.

Security analysis

As described above, the state space of a CPN comprises all the reachable states and occurrence sequences of transitions of the CPN. We can use the CPN Tools to generate the state spaces of the CPN models for SIP and the intruder, then we will be able to: 1) confirm known security threats and find out details of the threats; and 2) discover new security holes in SIP.

For 1) we firstly need define the desired security properties of SIP, in order to specify the abnormal states caused by security attacks. Then we search the state spaces for such abnormal states and the paths to the states. For example, in Fig. 2, the BYE request is for terminating an established SIP session. However, an attacker could send a forged BYE message to a SIP server to terminate a session (launch a BYE attack (Geneiatakis et al 2006)). So the security property of absence of BYE attack implies that the state spaces do not contain a state in which a BYE message is received by the server while the client is in its state for media transmission. To verify whether SIP satisfies this property, we search the state spaces for such a state by using CPN queries. Through observing the details of the states found and the paths leading towards such a state, we will gain insights into how such an attack could occur, and how we could possibly prevent it.

With 2), we will need to explore the state spaces for suspicious states. We will start from the state space report provided by the CPN Tools (Jensen et al 2007). From the report, we can obtain information on properties such as dead markings, dead transitions and boundedness properties. A dead marking corresponds to a terminal state of SIP (with intruder's presence). Thus we can check the details of the dead marking to see if it is a desirable terminal state of SIP. If not, it may indicate a suspicious state due to the intruder's operations. Then we can query the state space to see how this state is reached.

DISCUSSION AND FUTURE WORK

The Session Initiation Protocol is a core protocol of VoIP. In this paper we have discussed the security threats that SIP is facing and the importance of conducting formal security analysis of SIP specification. We have presented a Coloured Petri Net-based approach for assessing SIP security threats.

This approach is still in its early stage of development. The main issue the approach could face is the state space explosion problem. Therefore after we have done the complete modelling, we will investigate the severity of state space explosion problem and study the methods for alleviating the problem.

Ultimately, we will use the proposed methodology to carry out a rigorous and comprehensive security analysis of SIP, and use the CPN models created for SIP and intruders, to evaluate countermeasures proposed for securing SIP. This will be done by creating CPN models for the countermeasures and including them in the models for SIP and intruders, and then using the analysis techniques to assess those measures. We will also extend the approach presented in this paper to security analysis of other VoIP protocols.

REFERENCES

Abdelnur, H., Avanesov, T., Rusinowitch, M. and State, R. (2009) "Abusing SIP authentication". *J. of Information Assurance and Security*, vol. 4, pp. 311-318. Dynamic Publishers.

AI-Azzoni, I. Down, D.G. and Khedri, R. (2005) "Modelling and verification of cryptographic protocols using coloured Petri nets and Design/CPN". *Nordic J. of Computing*, vol. 12, no. 3, pp. 201-228

Billington, J., Gallasch, G.E. and Han, B. (2004) "Lectures on Concurrency and Petri Nets: A Coloured Petri Net Approach to Protocol Verification". *LNCS*, vol. 3098, pp. 210-290. Springer.

Dantu, R. Fahmy, S. Schulzrinne, H. and Cangussu, J (2009) "Issues and challenges in securing VoIP". *Computers & Security*, vol.28 , pp. 743-753. Elsevier.

Ding, L.G. and Liu, L. (2008) "Modelling and analysis of the INVITE transaction of the Session Initiation Protocol using coloured Petri nets". *LNCS* 5062, pp. 132-151. Springer.

Ding Y. and Su, G. (2007) "Intrusion detection for signal based SIP attacks through timed HCPN". *In Proc. of the 2nd Int. Conf. on Availability, Reliability and Security*. IEEE.

Ding, Y and Su, G. (2008) "A reduction method for verification of security protocol through CPN. In *Proc. of IEEE Int. Conf. on Networking, Sensing and Control*, pp. 73-77. IEEE.

Dolev, D. and Yao, A.C. (1983) "On the security of public key protocols". *IEEE Transactions on Information Theory*, vol. IT-29, no. 2, pp.198-208. IEEE.

Ehlert, S. Geneiatakis, D. and Magedanz, T. (2010) "Survey of network security systems to counter SIP-based denial-of-service attacks". *Computers & Security*, vol. 29, pp. 225-243. Elsevier.

Geneiatakis, D., et al (2006) "Survey of security vulnerabilities in session initiation protocol". *IEEE Communications Surveys & Tutorials*, vol. 8, No. 3, pp. 68-81. IEEE.

Jensen, K. (1997) *Coloured Petri nets: Basic concepts, analysis methods and practical use*, vol. 1, 2, and 3. 2nd ed. Springer.

Jensen, K. Kristensen, L. and Wells, L. (2007) "Coloured Petri nets and CPN tools for modelling and validation of concurrent systems". *Int. J. of Software Tools for Technology Transfer,* vol. 9, no. 3, pp. 213-254. Springer.

Murata, T. (1989) "Petri Nets: Properties, Analysis and Applications". *Proceedings of the IEEE*, 77(4):541-580. IEEE.

Nieh, B.B. and Tavares, S.E (1993) "Modelling and analyzing cryptographic protocols using Petri nets". In *Advances in Cryptology - AUSCRYPT'92, LNCS* 718, pp. 275-295. Springer.

Ormazabal, G., Nagpal, S., Yardeni, E. and Schulzrinne, H. (2008) "Secure SIP: A scalable prevention mechanism for DoS attacks on SIP based VoIP systems". *LNCS 5310*, pp. 107-132. Springer.

Permpoontanalarp, Y. and Sornkhom, P. (2009) "A new coloured Petri net methodology for the security analysis of cryptographic protocols". In *Proc. of the 10th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, pp. 59-78.

Rosenberg, J., et al. (2002). *RFC 3261: SIP: Session Initiation Protocol*. Internet Engineering Task Force.

Sengar, H., Wijesekera, D., Wang, H. and Jajodia, S. (2006) "VoIP Intrusion Detection Through Interacting Protocol State Machines". In *Proc. of the 2006 Int. Conf. on Dependable Systems and Networks*. IEEE.

Sisalem, D., et al. (2009) SIP Security. Wiley.

Sparks, R. (2007) "SIP: basics and beyond". *Queue*, vol. 5, no. 2, p. 22-33, ACM, New York.

The CPN Tools (2010) *Examples of Industrial Use of CP-nets*, retrieved on July 28, 2010 from http://www.daimi.au.dk/CPnets/intro/example_indu.html

VOIPSA (2010) *Home page of the Voice over IP Security Alliance (VOIPSA)*. Retrieved July 28, 2010 from http://www.voipsa.org/Activities/

Werapun, W., Kalam, A.A.E., Paillassa, B. and Fasson, J. (2009) “Solution analysis for SIP security threats”. In *Proc. of Int. Conf. on Multimedia Computing and Systems,* pp. 174-180,

Zhang, Z. (2007) “Security analysis of session initiation protocol”. *Int. J. of Innovative Computing, Information and Control*, vol.3, no. 2. ICIC International.