International Journal of Information & Network Security (IJINS)
Vol.2, No.5, October 2013, pp. 403~416
ISSN: 2089-3299

Modeling Kerberos Authentication Protocol Using Colored Petri Net

Amir Keshvari Ilkhichi*, Saeid Pashazadeh**
* Faculty of Electrical, Computer & IT Engineering, Islamic Azad University, Zanjan Branch
** Faculty of Electrical and Computer Engineering, University of Tabriz

Article Info

Article history:

Received Jun 14th, 2013
Revised Aug 20th, 2013
Accepted Sep 16th, 2013

Keyword:

Modeling
Verification
Colored Petri Net
Authentication Protocol
Kerberos

ABSTRACT

Information security is essential in today's digital world and plays an important role in message exchanges and trading. Authentication protocols play an important role in information security. Kerberos is one of the famous and commonly used authentication protocols. These factors cause attention of many researchers for formal modeling and analyzing of security properties of this protocol. Colored Petri net is powerful formal modeling language with great modeling capabilities and is one of the suitable methods for verifying properties of various systems like security protocols. Hierarchical modeling of Kerberos authentication protocol version 5 using colored Petri net is presented in this paper. Operations and messages of Kerberos protocol are modeled with full detailes. Presented model can be extended easily for studying different properties of this protocol and its strength against various attacks.

Corresponding Author:

Second Author,
Department of Information Technology
Faculty of Electrical and Computer Engineering,
University of Tabriz,
29th Bahman Boulevard, Tabriz, East Azerbaijan Province, Iran.
Email: pashazadeh@tabrizu.ac.ir

1. INTRODUCTION

Growth and wide spread use of computer networks and particularly the Internet in many activities of organizations and institutions makes widespread changes in human's lifestyles. Information security is an essential and important issue in this area. Connecting local network of an organization to global networks places data of organizations in risk of exposed access by external hosts. Confidentiality of sensitive information from unauthorized access on the internet is the most important security challenges. Authentication is one of the basic mechanisms for enforcing confidentiality. Authentication is process of identifying and ensureing for identity of a party for enforcing confidentiality preventing impersonation. Authentication is one of the key fields of security in the information exchange network. Authentication verifies the identity of every user who wants to access network's resources.

Kerberos is an authentication service developed as part of Athena project at MIT and is based on Needham–Schroeder protocol [1]. Kerberos belongs to the mid-eighties, but its new applications became a reality in the modern operating systems like Linux, Windows, and Solaris. It works based on tickets and allows nodes communicating over an insecure network to prove safely their identity to one another. It is designed based on client-server model and built on symmetric key encryption. Kerberos supports mutual authentication and both of the client and the server prove their identities to each other. Some important feautures of Kerberos are:

- **Security**: A network eavesdropper is not able to obtain the necessary information to impersonate a user.

- **Reliability**: For all services that rely on Kerberos for access control, lack of availability of the Kerberos service means lack of availability of the supported services.
- **Transparency**: Ideally, the user should not be aware that authentication is taking place, beyond the requirement to enter a password.
- **Scalability**: The system should be capable of supporting large numbers of clients and servers.

2. Related Works

Widespread use of Kerberos as one of the important and widespread protocols of network autheticaton requires formal modeling and verification of it. Many modeling languages have been introduced, but only those are suitable for automated analysis that have mathematical basic and have formalism. A formalization of Kerberos 4 is presented in [2] and extended analyzis of it is done using an inductive approach [3],[4],[5],[6]. Isabelle theorem prover is used in some of formal analysis of Kerberos and yielded formal correctness proofs for a specification with timestamps, and also highlighted a few minor problems of this protocol. Kerberos version 5 is studied using Murφ state exploration tool [7]. Verification and formalization of Kerberos protocol using NuSMV model checker is also studied [8]. Some improvements are proposed for preventing replay attacks and password attacks using triple password scheme [9]. Adyanthaya et. Al [10] has used NuSMV tool for studying basic version of Kerberos for authentication. They have presented the concept of freshness which helps for finding scenarios of reply attack. They demonstrated a possible means by which the weakness of the Kerberos protocol causing success of replay attack. They made suggestions for making the protocol more stronger. The modelling of the basic authentication procedure presented in [11]. In this paper shows that security increases with attributes quantity and decreases with possibility of repeating wrong sequence of symbols. Security analysis of the Kerberos protocol using BAN logic is proposed in [12]. Reliability, practicability and security of Kerberos protocol are proved in this paper. CSP methods is used for formalization of Kerberos protocol [13]. Security of the Kerberos protocol is proved in the way of formal methods in this paper.

Using colored petri net as formal method for verification of security protocols is done in [14]. Coloured Petri net is a formal modeling language that benefits from graphical modeling. It is used for modeling and verification of wide range of topics in compuer science [15]. Great ability of it for model concurrent systems and its flexibility makes it an appropriate tool for analysis of wide range of network and security protocols. Colored petri is extension of classical Petri net by allowing to define color for tokens and having various type of tokens [16], [17]. Coloured Petri nets benefits from a powerful functional ML language that is an artificial intelligence language that is based on the lambda-calculus for defining colors and inscriptions [18]. Other formal tools also is used for some properties of new version 5 of Kerberos [19].

In this paper, formal modeling of Kerberos version 5 using colored petri net is presented. For this purpose, full schema of protocol dialogue is presented in next section and formal modeling of its operation is modeled in future sections.

3. PROTOCOL DIALOGUE

Full decsreption of Kerberos protocol is presented in most textbooks of computer and internet security [1]. We assumed that readers are familiar with kerberos protocol. Figure 1 shows summary of protocol's dialogue. Four basic entities that play major rules in Kerberos protocol Version 5 are *Authentication server (AS)*, *Ticket granting server (TGS)*, *Server(S)*, *Client(C)*. Table 1 shows summary of message exchange in Kerberos authentication protocol version 5. Some notations are used in the protocol that is introduced in this part of paper. $K_{a,b}$ represents symmetric (shared) session key between parties a and b. E(K, [X]) represents symmetric encryption of plaintext X using secret key K. x || y represents x concatenated with y. K_v represents secret symmetric key shared by authentication server and party V.

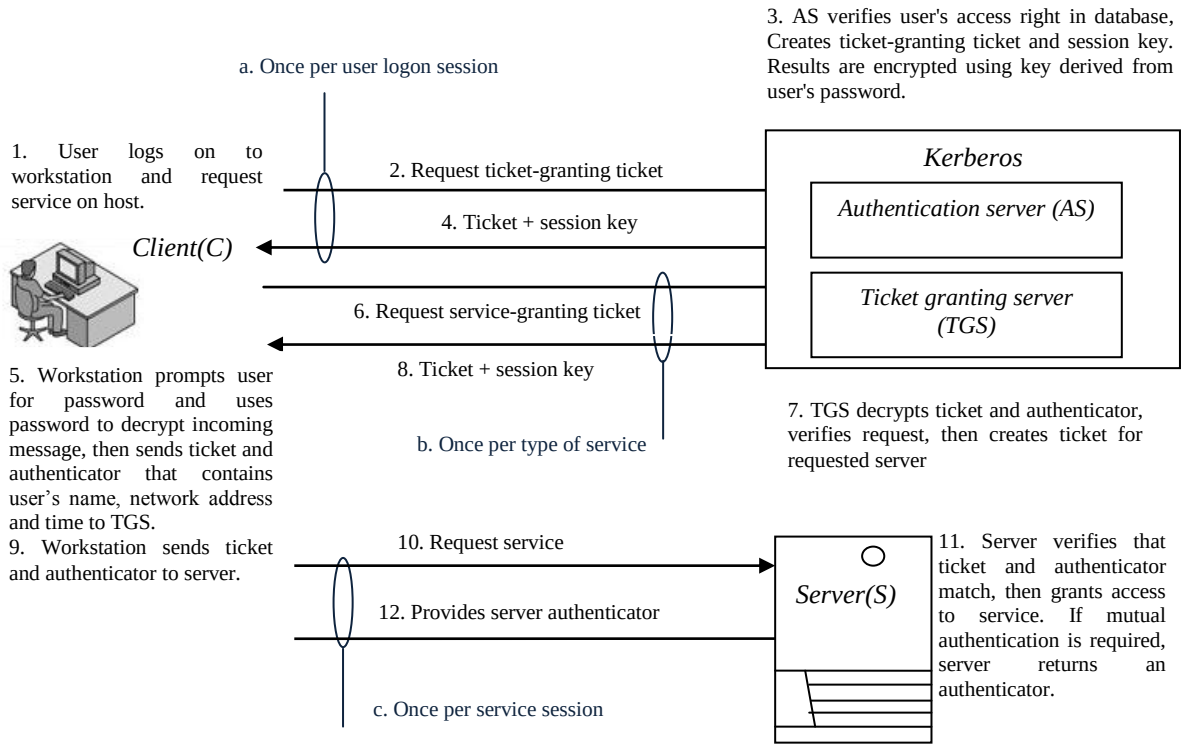

Figure 1. Dialogue of Kerberos protocol.

Table 1. Message exchanges of Kerberos version 5.

(1) **C → AS** $\text{Options} \| \text{ID}_c \| \text{Realm}_c \| \text{ID}_{tgs} \| \text{Times} \| \text{Nonce}_1$

(2) **AS → C** $\text{Realm}_c \| \text{ID}_c \| \text{Ticket}_{tgs} \| E(K_c, [\, K_{c,tgs} \| \text{Times} \| \text{Nonce}_1 \| \text{Realm}_{tgs} \| \text{ID}_{tgs}])$

$\text{Ticket}_{tgs} = E(K_{tgs}, [\text{Flags} \| \ K_{c,tgs} \| \text{Realm}_c \| \text{ID}_c \| \ \text{AD}_c \| \text{Times}])$

(a) Authentication Service Exchange to obtain ticket-granting ticket

(3) **C → TGS** $\text{Options} \| \text{ID}_v \| \text{Times} \| \text{Nonce}_2 \| \text{Ticket}_{tgs} \| \text{Authenticator}_c$

(4) **TGS → C** $\text{Realm}_c \| \text{ID}_c \| \text{Ticket}_v \| E(K_{c,tgs}, [\, K_{c,v} \| \text{Times} \| \text{Nonce}_2 \| \text{Realm}_v \| \text{ID}_v])$

$\text{Ticket}_{tgs} = E(K_{tgs}, [\text{Flags} \| \ K_{c,tgs} \| \text{Realm}_c \| \text{ID}_c \| \text{AD}_c \| \text{Times}])$
$\text{Ticket}_v = E(K_v, [\text{Flags} \| \ K_{c,v} \| \text{Realm}_c \| \text{ID}_c \| \text{AD}_c \| \text{Times}])$
$\text{Authenticator}_c = E(K_{c,tgs}, [\, \text{ID}_c \| \text{Realm}_c \| \text{TS}_1])$

(b) Ticket-Granting service Exchange to obtain service-granting ticket

(5) **C → V** $\text{Options} \| \text{Ticket}_v \| \text{Authenticator}_c$

(6) **V → C** $E(K_{c,v}, [\, \text{TS}_2 \| \text{Subkey} \| \text{Seq\#}])$

$\text{Ticket}_v = E(K_v, [\text{Flags} \| \ K_{c,v} \| \text{Realm}_c \| \text{ID}_c \| \text{AD}_c \| \text{Times}])$
$\text{Authenticator}_c = E(K_{c,v}, [\, \text{ID}_c \| \text{Realm}_c \| \text{TS}_2 \| \text{Subkey} \| \text{Seq\#}])$

(c) Client/Server Authentication Exchange to obtain service

4. COLOR SETS, INITIAL MARKINGS AND MODEL OF KERBEROS SYSTEM

4.1. Color Sets

Declaration of color sets that are used in modeling of protocol are as follows:

```
colset ID = with IDc | IDtgs | IDv;
colset Realm = with Realmc | Realmtgs | Realmv;
colset OPTIONS = with Options | Flags;
colset AD =STRING;
colset T = with From | Till | Rtime;
colset Times = record from:T * till:T * rtime:T;
colset TS12 = with Date1 | Time1 | Date2 | Time2;
colset TS = record d:TS12 * t:TS12;
colset NONCE = INT;
colset M1 = record options:OPTIONS * idc:ID * realmc:Realm * idtgs:ID * times:Times *
                   nonce1:INT;
colset Key = with Kc | Kctgs | Kastgs | Kcv | Ktgsv | Subkey;
colset Ttgs = record kastgs:Key * flags:OPTIONS * kctgs:Key * realmc:Realm * idc:ID * adc:AD *
                   times:Times;
colset CTM2 = record kc:Key * kctgs:Key * times:Times * nonce1:INT * realmtgs:Realm *
                    Idtgs: ID;
colset M2 = record realmc:Realm * idc:ID * ttgs:Ttgs * ctm2:CTM2;
colset Ac3 = record kctgs:Key * idc:ID * realmc:Realm * ts1:TS;
colset M3 = record options:OPTIONS * idv:ID * times:Times * nonce2:INT * ttgs:Ttgs * ac:Ac3;
colset Tv = record ktgsv:Key * flags:OPTIONS * kcv:Key * realmc:Realm * idc:ID * adc:AD *
                    times:Times;
colset CTM4 = record kctgs:Key * kcv:Key * times:Times * nonce2:INT * realmv:Realm * idv:ID;
colset M4 = record realmc:Realm * idc:ID * tv:Tv * ctm4:CTM4;
colset Ac5 = record kcv:Key * idc:ID * realmc:Realm * ts2:TS * subkey:Key * seq:INT;
colset M5 = record options:OPTIONS * tv:Tv * ac:Ac5;
colset M6 = record kcv:Key * ts2:TS * subkey:Key * seq:INT;
```

Color set ID is defined to represent identifier of C user, TGS server and V server. Computer networks of clients and servers under different administrative organizations typically constitute different realms. Color set Realm represents the realm of C, TGS and V. Color set OPTIONS represents Status of Tickets. Color set AD is defined to represent network address of the user. Color set T is defined as enumerated type for representing Time with three different values. Color set Times is defined to represent the desired start time for the requested ticket, requested expiration time for the ticket, and requested renew-till time of it. Color set TS12 represents date and time with different values. Color set TS is used to represent date and time of sending the Authenticator_c in message 3 and 5. Color set NONCE is defined to represent a random number that is used in Kerberos protocol. Color set M1 is defined to represent Message 1 of kerberos Protocol Version 5. Color Set Key is used to display all the keys used in the protocol. Table 2 shows enumerated values that are used in defining color set Key and berief description of them. These values will be used in CPN model of Kerberos protocol in following figures.

Table 2. Keys that are used in Kerberos V5.

key	Description
Kc	User secret key
Kctgs	Session key between user and TGS server
Kastgs	Secret key between AS server and TGS server
Kcv	Session key between user and V server
Ktgsv	Secret key between TGS server and V server
Subkey	Encryption key to be used to protect this specific application session

Color set Ttgs is defined to represent color set of the ticket generated by the AS server. In addition, the Color set Ttgs is encrypted by the key Kastgs. Color set CTM2 is used to display text encrypted by the user encryption key in a message 2. Color set M2 is defined to represent message 2 of protocol. Color set Ac3 is used for representing type of Authenticator_c in message 3. It is encrypted by the key Kctgs. Authenticator_c is generated by client to validate ticket. Color set M3 is defined to represent message 3 of Kerberos protocol. Color set Tv is defined to be used as type of ticket that is generated by the TGS server. Color set Tv is encrypted by the key Ktgsv. Color set CTM4 is used to display text encrypted by the Kctgs encryption key in a message 4. Color set M4 is defined to represent message 4 of kerberos protocol. Color set Ac5 is used to display Authenticator_c in message 5 of protocol. It is encrypted by the key Kcv. Authenticator_c is generated by client to validate ticket. Color set M5 is defined to represent message 5 of Kerberos protocol. Color set M6 is defined to represent message 6 of protocol and is encrypted by the key Kcv.

4.2. Initial Markings and Variables

Initial markings are assignment of tokens to places in initial state of model that is visible in all figures with green color. Table 3 shows variables that are used in models of system.

4.3. Hierarchical Model of Kerberos V5

CPN tool supports modular and hierarchical modeling. A hierarchical model of Kerberos protocol version 5 is presented in this paper. Figure 2 shows top-level model of protocol. This model contains four basic sub-modules. Four substitution transitions of top-level model refer to these four following sub-modules.

1. C substitution transition refers to *Client* sub-module that models user entity.
2. AS substitution transition refers to *Authentication server* sub-module that models authentication server entity.
3. TGS substitution transition refers to *Tikect granting serevr* sub-module that models ticket granting server entity.
4. V substitution transition refers to *Server* sub-module that models server entity.

Colored Petri net model of protocol is completely based on the protocol's dialogue that is presented in Figure 1 and Table 1. In Figure 2, client sends a message m1 to the AS and requests access to the TGS. The AS responds to client with a message m2 that is encrypted with a key derived from the user's password (Kc) that contains the ticket and session key $K_{c,tgs}$ where the subscripts indicate that this is a session key for C and TGS. Then C sends a message to the TGS that includes the ticket plus the ID of the requested service (message m3). Reply message (m4) from the TGS follows the form of message (m2). The message is encrypted with the session key shared by the TGS and C and includes a session key to be shared between C and the server V. C now sends a message m5 to the V. Server replays with a message m6 for mutual authentication. Only substitution transition C in Figure 2 is enabled at start of simulation run.

Table 3. Variables of CPN model

VARIABLES	
Var m1 : M1;	Var ac5 : Ac5;
Var i1,i2 : ID;	Var m5 : M5;
Var r : Realm;	Var m6 : M6;
Var k1,k2 : Key;	Var t1,t2,t3 : T;
Var option : OPTIONS;	Var times1 : Times;
Var m2 : M2;	Var expirationtimes : BOOL;
Var ttgs1 : Ttgs;	Var t4,t5 : TS12;
Var ctm21 : CTM2;	Var ts1,ts2 : TS;
Var ac1 : Ac3;	Var nonce1,nonce2,nonceorg : NONCE;
Var m3 : M3;	Var b : BOOL;
Var tv1 : Tv;	Var ad : AD;
Var ctm41 : CTM4;	Var seqnum : INT;
Var m4 : M4;	

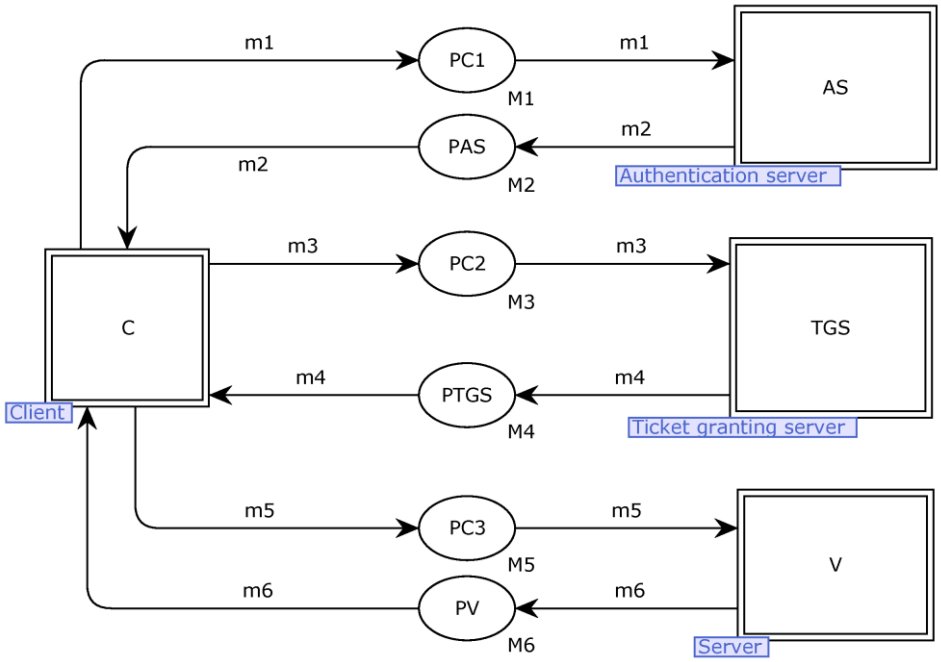

Figure 2. Top level CPN model of Kerberos system version 5.

CPN models of four basic sub-modules of protocol are presented in this part of paper. Figure 3 shows CPN model of *Client* sub-module. This module consists of four separate sub-nets that each sub-net represents a specific task that client entity palys as its rule in the Kerberos protocol. First sub-net that is shown in top of Figure 3 models initiating a message exchange. This sub-net models generation of first message of protocol by *Client* that is shown as first message in Table 1. Client sends this message to *Authentication server* for requesting access to the TGS. Client entity generates a token of type M1 in this step of protocol and sends it to authentication server. At start simulation run, transition TC101 is enabled as is shown in Figure 3. By firing of transition TC101, fields of Times and Nonce1 in message 1 of protocol is generated. Firing of transition TC102 generates all fields of message 1 and puts a token of type M1 in output place PC109. The Nonce1 is a random value that is generated by client and will be repeated in second message of protocol (as is shown in Table 1) and will come back to client from authentication server to assure that the response is fresh and has not been replayed by an opponent. In Figure 3, function rand() generates Nonce1 value and puts it as a token in place PC105.

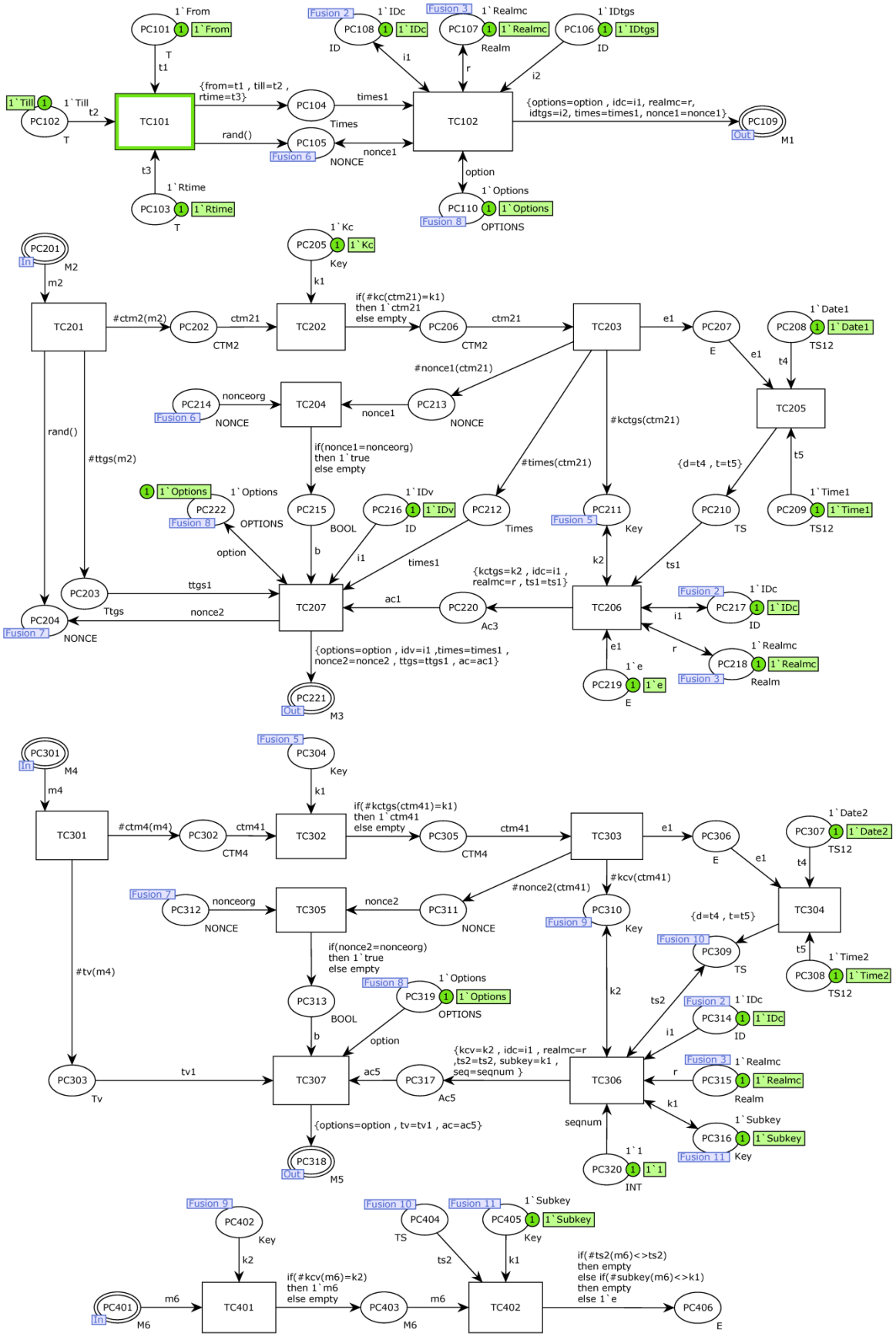

Figure 3. CPN model of client sub-module.

Nine different fusion sets are used in this module of Figure 3 to control execution order of subnet's transitions and checking validity of messages and reducing drawing long crossing arcs. For instance, *Client* must remember Nonce value that it randomly generated in the first subnet, in order to verify validity of response that it receives in second subnet.

In second subnet of Figure 3, client takes a token of type M2 from input place PC201 and puts a token of type M3 in output place PC221. This sub-net models operations of *Client* entity for (1) decomposition of recived message from authentication server that is shown as second message of protocol in Table 1. (2) Generation of new message that is shown as third message of protocol that will be sent to ticket granting server. In this subnet, the received cipher text of message 2 will be decrypted by firing of transition TC202. Client decrypts cipher text by key (kc) to access the session key, $K_{c,tgs}$, that this is sent by AS for using as session key between C and TGS. Firing of transition TC203 puts the session key ($K_{c,tgs}$) in place PC211. The field Authenticator_c of message 3 is generated by firing of transition TC206 and is placed in place PC220. Authenticator contains the ID and realm of client user and a timestamp. Unlike the ticket that is reusable, the authenticator is intended for being used only once and has a very short lifetime. Client checks received message by comparing the Nonce that client had been choosen and value of Nonce that is in the received message 2. Finally, firing of transition TC207 generates message 3.

Third subnet of Figure 3 is similar to the second subnet. Client entity takes a token of type M4 from input place PC301 and puts a token of type M5 in output place PC318. This sub-net models operation of client entity for (1) decomposition of recipted message from ticket granting server that is shown as forth message of protocol's messages in Table 1 and (2) generation of new message that is shown as fifth message of protocol in Table 1 that will be sent to server. Received cipher text of message 4 will be decrypted by firing of transition TC302 in this subnet. Client decrypts cipher text using the key ($K_{c,tgs}$) to access the session key ($K_{c,v}$) t is session key between C and V. Firing of transition TC303 puts the session key ($K_{c,v}$) in place PC310. Authenticator_c field of message 5 is generated by firing of transition TC306 and is placed in place PC317. Client authenticates received message by comparing the Nonce that client had been choosen and value of Nonce that is repeated in the received message 4. Finally, by firing of transition TC307 message 5 will be generated.

In fourth subnet of Figure 3, client takes a token of type M6 from input place PC401. By firing of transition TC401, client decrypts this message and recovers the subkey and timestamp. Because the message is encrypted with the session key, client will be assured that it had been been created only by server (mutual authentication). Client and server share a secret key at end of process.

Figure 4 shows CPN model of authentication server sub-module. In this submodule, operations of authentication server for (1) reception of first message of protocol that is shown in Table 1, (2) decomposition of this message, and (3) generation of response message that is shows as second message of protocol and sending it to the client is modeled. After receiption of first message from client, authentication server generates a ticket for access of client to ticket granting server and sends second message of protocol to the client. Authentication server takes a token of type M1 from input place PAS01. Firing of transition TAS03 generates ciphertext of second message and holds it in place PAS10. Firing of transition TAS02 generates ticket and plain text of second message and sends them to places PAS06. Place PAS07 contains user's password (K_c) and place PAS08 contains the session key ($K_{c,tgs}$) that is generated by AS. Session key is placed inside the message encrypted with K_c and causes that only client can read it. The same session key is included in the ticket, which can be read only by the TGS. Thus, session key securely will be delivered to both C and the TGS. Firing of transition TAS05 puts a token of type M2 in output place PAS11. This corresponds to the second message of protocol that authentication server sends to client.

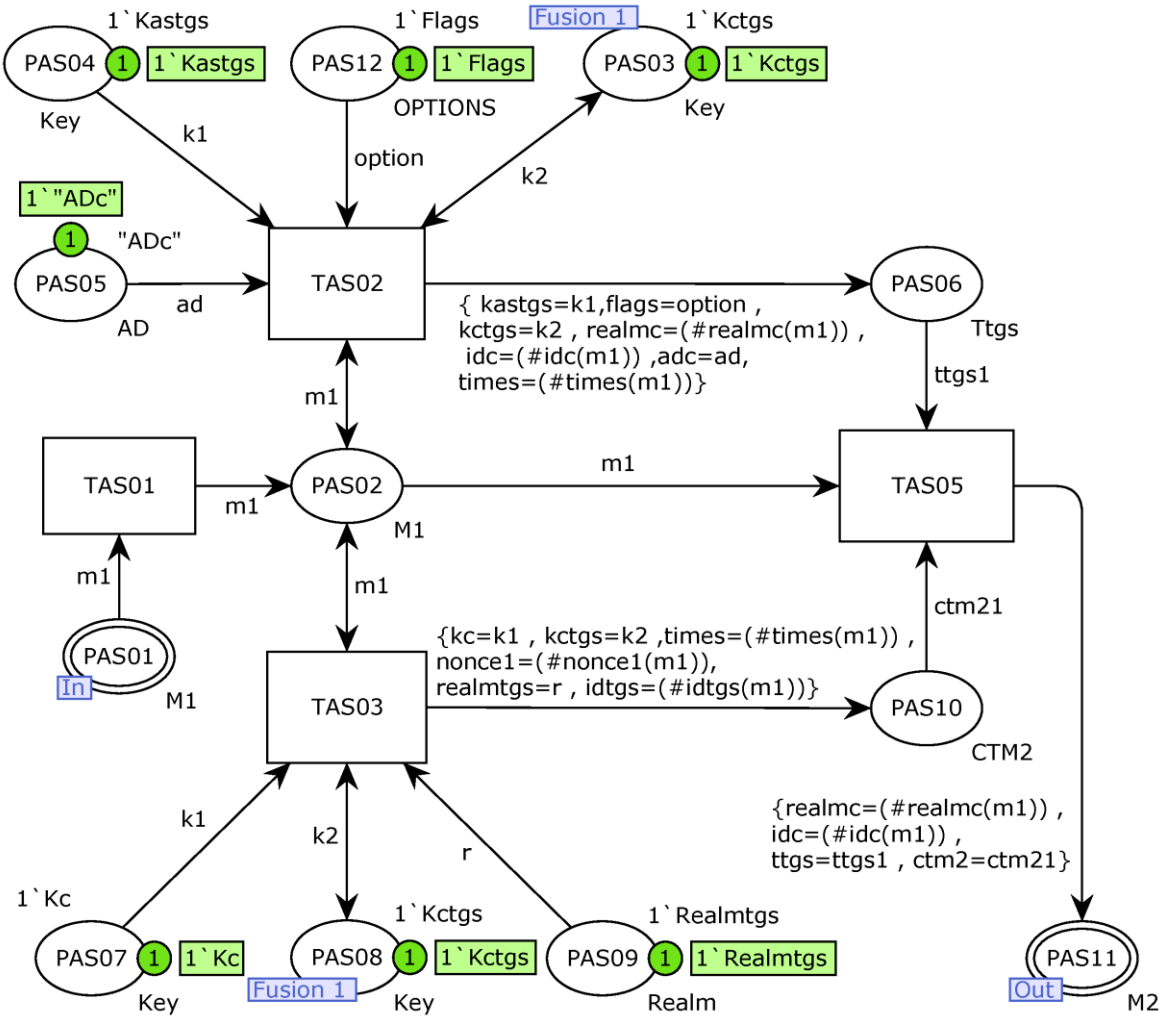

Figure 4. CPN model of authentication server sub-module.

Figure 5 shows the CPN model of ticket granting server sub-module. This sub-module represent operations of ticket granting server for (1) reception of third message of protocol that is shown in Table 1, (2) decomposition of this message, and (3) generation of response message that is shows as fourth message of protocol and sending it to the client. TGS sends the ticket for client. Ticket permits the client to use service of server. TGS decrypts $Authenticator_c$, $Ticket_{tgs}$ and checks ticket to ensure that non-impersonating agent has sent the ticket for cilent to access server. TGS takes a token of type M3 from input place PTGS01. By firing of transition TTGS02, it decrypts the received encrypted ticket of message 3 with session key $K_{as,tgs}$ that is shares between TGS and AS. This ticket indicates that user C has been provided with the session key $K_{c,tgs}$ by AS. The TGS uses the session key to decrypt the authenticator. TGS compares the ID and realm of the authenticator with that of the ticket and with the network address (AD_c) of the incoming message. If all of them match, then TGS assures that the sender of the ticket is real owner of it. Ticket does not prove anyone's identity but is a way to distribute keys securely. Authenticator proves the client's identity. Authenticator can be used only once, so it has short lifetime.

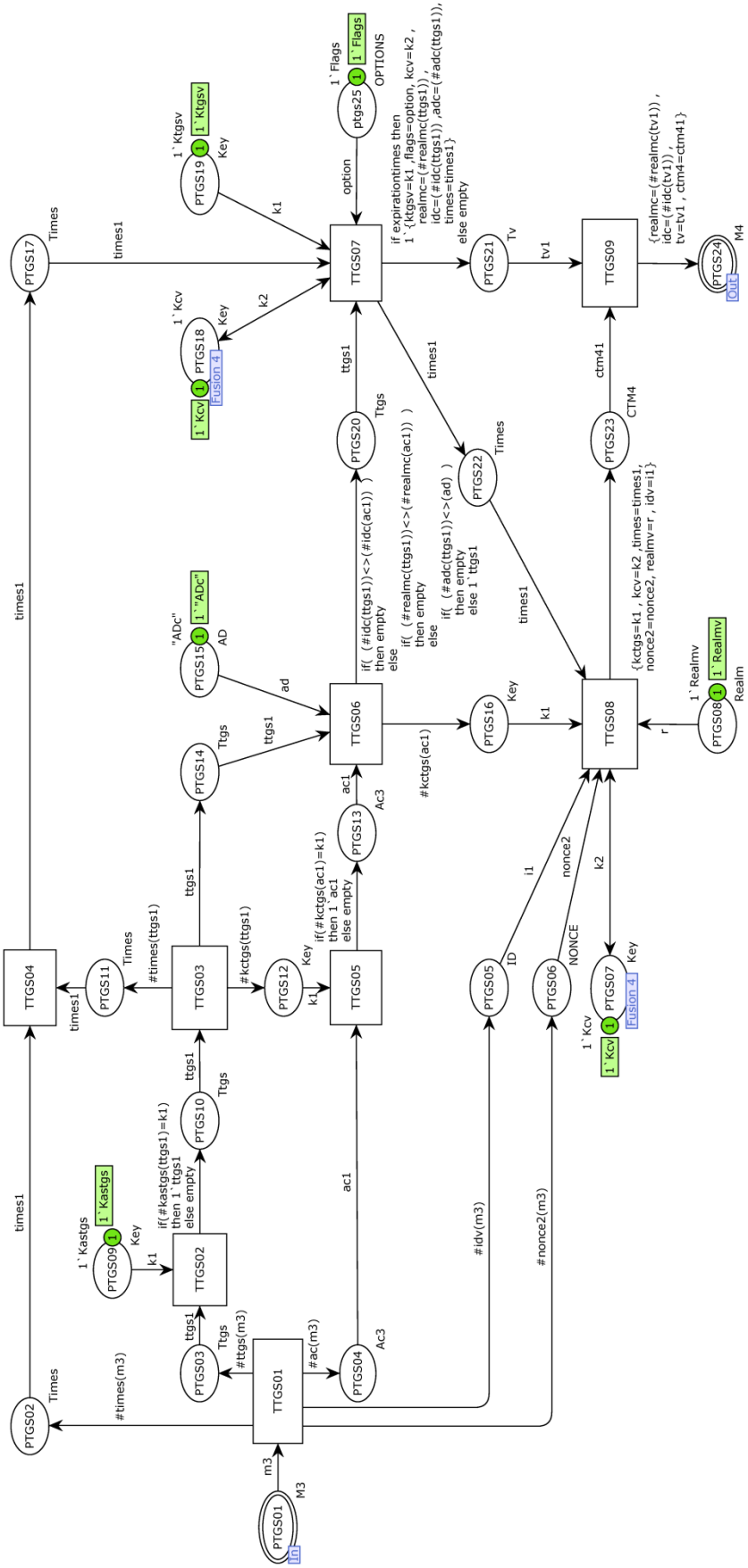

Figure 5. CPN model of ticket granting system sub-module.

Session key $K_{c,tgs}$ will be extracted after decryption of the ticket by firing of transition TTGS03. $Authenticator_c$ is decrypted by firing of transition TTGS05 with session key $K_{c,tgs}$. By firing of transition TTGS06, fields $Realm_c \parallel ID_c \parallel AD_c$ in ticket with fields $Realm_c \parallel ID_c \parallel AD_c$ in $Authenticator_c$ is compared. If they were equal, indicates validation of message sender's identity and then token is put in a place. Ticket of message 4 is generated by firing of transition TTGS07 and is placed in PTGS21. Ciphertext of message 4 is generated by firing of transition TTGS08 and is stored in place PTGS23. Cipher text is encrypted with session key ($K_{c,tgs}$) that is located in place PTGS16. The cipher text contains the key ($K_{c,v}$) that is generated by TGS and is hold in place PTGS07. Session key is inside the message that is encrypted with $K_{c,tgs}$, so only the client can read it. The same session key is included in the ticket, which can be read only by the server. Thus, the session key securely will be delivered to both of C and the V. Finally, a token of type M4 is sent to output place PTGS24 by firing of transition TTGS09. This corresponds to the fourth message of the protocol.

Figure 6 shows CPN model of server sub-module. If mutual authentication is required, the server can reply as shown in message (6) of Table 1. This sub-net models operation of server for (1) decomposition of recipted message from client that is shown as fifth message of protocol in Table 1 and (2) generation of new message that is shown as sixth message of protocol that will be sent to client. Server decrypts $Authenticator_c$ and $Ticket_v$. It compares user ID and realm in ticket with user ID and realm in $Authenticator_c$ and similarly the user's network address to become sure that no user impersonation occurred. Then server sends a message that contains seq field of $Authenticator_c$ in received message number 5 to client for confirming the identity of server. Server takes a token of type M5 from input place PV01. It decrypts the received encrypted ticket of message 5 with session key $K_{tgs,v}$ by firing of transition TV02. Session key $K_{c,v}$ will be extracted after decryption of ticket by firing of transition TV03. By firing of transition TV04, $Authenticator_c$ is decrypted with session key $K_{c,v}$. $Realm_c \parallel ID_c \parallel AD_c$ fields in ticket with fields $Realm_c \parallel ID_c \parallel AD_c$ in $Authenticator_c$ are compared by firing of transition TV05. If they were equal, indicates validation of message sender's identity and then a token is put in place PV10. Message 6 is generated and is placed in place PV11 by firing of transition TV06.

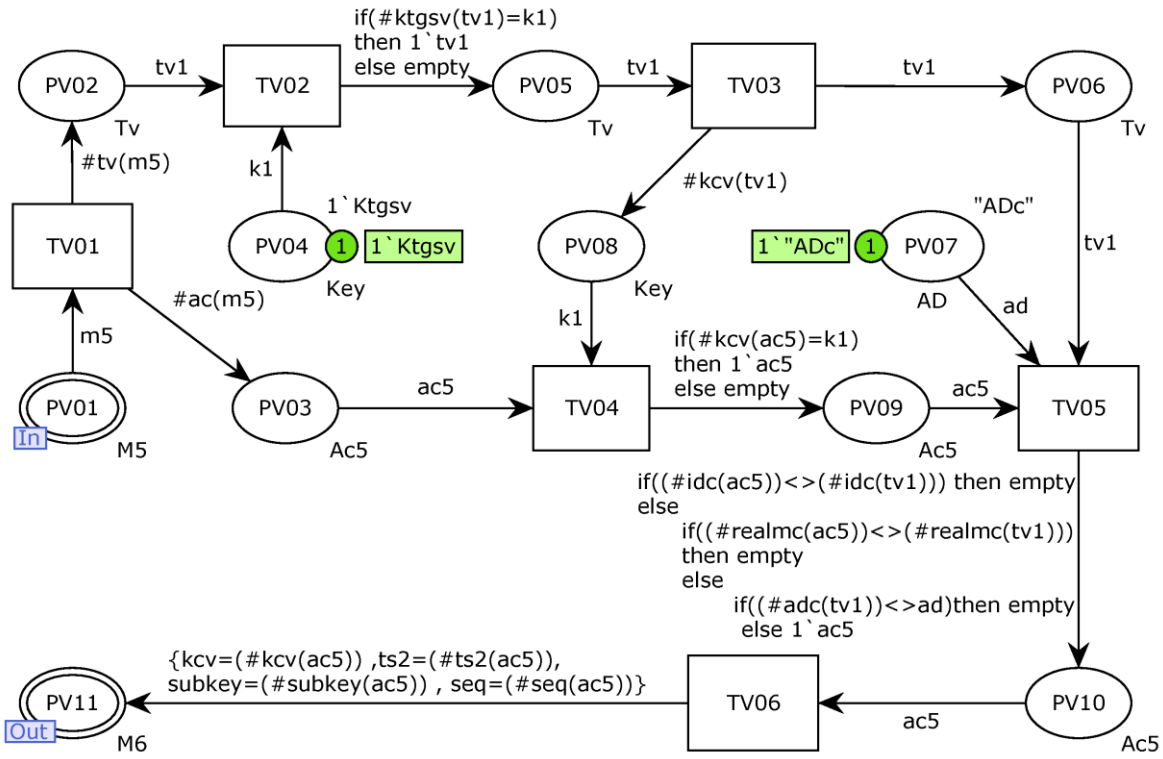

Figure 6. CPN model of server sub-module.

5. FUNCTIONS AND ARC EXPRESSION OF MODEL

Function rand is defined as follows to generate a random number that will be used as Nonce field of messages 2, and 3 of the protocol.

```
fun rand() = discrete(100,999);
```

Following arc expressions is used in model to compare the value of the Nonce value in message 2 to ensure that the response is fresh and has not been responded to by opponent.

```
If (nonce1 = nonceorg) then 1`true
else empty
```

In following arc expressions, the TGS server compares ID and realm and the network address of user in Ticket_{tgs} and Authenticator_{c}. If all the fields are matched, the server ensures that the sender of the ticket is actual owner of the ticket.

```
If ((#idc(ttgs1)) <> (#idc(ac1))) then empty
else
    if ((#realmc(ttgs1)) <> (#realmc(ac1))) then empty
    else
        if ((#adc(ttgs1)) <> (ad))  then empty
        else 1`ttgs1
```

In following arc expressions, if the value of expirationtimes field is true then allows sending ticket, and in otherwise, the server does not send the ticket to user from server. Expirationtimes field indicates the validity period of the ticket.

```
if expirationtimes then
    1`{ktgsv=k1, flags=option, kcv=k2 , realmc=(#realmc(ttgs1)),
        idc=(#idc(ttgs1)) ,adc=(#adc(ttgs1)), times=times1}
else empty
```

6. STATE SPACE OF SYSTEM

Part of state space report of model is as follows:

```
Statistics                          Home Properties
  ------------------------------------------------------------------------------------------------
State Space                         Home Markings
   Nodes:  47                           None
   Arcs:   53                       Liveness Properties
   Secs:   1                        ----------------------------------------------------
   Status: Full                     Dead Markings
 Scc Graph                              [29,47]
   Nodes:  47                       Dead Transition Instances :
   Arcs:   53                           None
   Secs:   0                        Live Transition Instances
                                        None
                                    Fairness Properties
                                    ----------------------------------------------------
                                    No infinite occurrence sequences
```

This report is result of single run of the protocol for one time access of client for a server. Dead markings of system shows one time run of protocol. This report shows that protocol do not have deadlock state. Modeling of introduer and studying successful scenarios of attacks are under study.

7. CONCLUSION

Authentication protocols plays important rule in computer security. Kerberos is one of the most frequently used protocols of network authentication. Kerberos version 5 is lastes released version of this

protocol up to time of publication of this paper that was extended Kerberos version 4 and was improved for increasing its security. Formal modeling and analysis of such important and widely used protocol is required. Most of these protocols suffer from leakage of formal verification of their security properties and strength against specific attacks. Different tools for modeling and formal verification of systems and protocols are developed. Some of these tools are appropriate for analyzing security protocols. General modeling and verification tools have some benefits and drawbacks against specialized modeling tools. General modeling tools has benefit that permits modeler to models wide range of protocols and attackes in comparision with specialized modeling tools. Modeling some attacks is very simple and can be done in short time using specialized modeling tools in comparision with general tools. Colored petri net is used as general modeling and verification tool. Colored Perti net has great flexibility for modeling wide range of security protocols and attacks. The price we pay for flexibility is bigger size of models of security protocols and attacks against them.

In this paper, a hierarchical colored Petri net model of Kerberos version 5 is presented. Some minor details of protocol are eliminated in modeling but all major parts of protocol are modeled. Model contains four sub modules for clinet, authentication server, server, and ticket granting server. Main aim of paper is modeling of Kerberos version 5. Based on experiments of authors, modeling of protocol is straightforward work although figures of model seem alittle complex in first glance. Running of model in simulation mode clearly visualizes steps of protocol and helps deeper understanding the operations of kerberos. State space analysis of model shows that models runs without deadlock. Main contribution of the paper is that modeling of Kerberos Version 5 using colored Petri net is done for first time in this paper. Using features of ML language of CPN Tool for model checking of state space of model gives capability of verifying properties of Kerberos protocol. This basic model can be extended for future studies such as analyzing strength of protocols against various attacks. State space analysis using model checking, helps us in finding scenarios of successful attacks against Kerberos. Modeling and study of man-in-the-middle attack with extending this model by adding an intruder is under study.

REFERENCES

[1] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 4th ed. Upper Saddle River, New Jersey, USA: Pearson/Prentice Hall, 2006.

[2] G. Bella and E. Riccobene, "Formal Analysis of the Kerberos Authentication System," *Journal of Universal Computer Science*, vol. 3, pp. 1337-1381, 1997.

[3] G. Bella, *Formal Correctness of Security Protocols*. Heidelberg, Germany: Springer-Verlag Berlin Heidelberg, 2007.

[4] G. Bella and L. C. Paulson, "Using Isabelle to Prove Properties of the Kerberos Authentication System," presented at the DIMACS Workshop on Design and Formal Verification of Security Protocols, Piscataway, NJ, USA, 1997.

[5] G. Bella and L. Paulson, "Kerberos Version IV: Inductive Analysis of the Secrecy Goals," in *Computer Security — ESORICS 98*. vol. 1485, J.-J. Quisquater, Y. Deswarte, C. Meadows, and D. Gollmann, Eds., ed Heidelberg, Germany: Springer, 1998, pp. 361-375.

[6] G. Bella and L. Paulson, "Mechanising BAN Kerberos by The Inductive Method," in *Computer Aided Verification*. vol. 1427, A. Hu and M. Vardi, Eds., ed Heidelberg, Germany: Springer 1998, pp. 416-427.

[7] J. C. Mitchell, M. Mitchell, and U. Stern, "Automated Analysis of Cryptographic Protocols Using Mur&phi," in *1997 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 1997, pp. 141-151.

[8] P. Mundra, S. Shukla, M. Sharma, R. M. Pai, and S. Singh, "Modeling and Verification of Kerberos Protocol Using Symbolic Model Verifier," in *2011 International Conference on Communication Systems and Network Technologies (CSNT)*, Katra, Jammu, India, 2011, pp. 651-654.

[9] G. Dua, N. Gautam, D. Sharma, and A. Arora, "Replay Attack Prevention in Kerberos Authentication Protocol Using Triple Password," *International Journal of Computer Networks & Communications (IJCNC)* vol. 5, pp. 59-70, March 2013.

[10] S. Adyanthaya, S. Rukmangada, A. Tiwari, and S. Singh, "Modeling Freshness Concept to Overcome Replay Attack in Kerberos Protocol Using NuSMV," in *2010 International Conference on Computer and Communication Technology (ICCCT)*, Allahabad, Uttar Pradesh, India, 2010, pp. 125-129.

[11] J. Capek, M. Hub, and R. Myskova, "Basic Authentication Procedure Modelled by Petri Nets," *International Journal of Computers and Communications* vol. 4, pp. 101-108, 2010.

[12] F. Kai, L. Hui, and W. Yue, "Security Analysis of the Kerberos Protocol Using BAN Logic," in *Fifth International Conference on Information Assurance and Security (IAS '09)* Xi'an, China, 2009, pp. 467-470.

[13] L. Qin, Y. Fan, Z. Huibiao, and Z. Longfei, "Formal Modeling and Analyzing Kerberos Protocol," in *2009 WRI World Congress on Computer Science and Information Engineering*, Los Angeles, CA , USA, 2009, pp. 813-819.

[14] I. Al-Azzoni, D. G. Down, and R. Khedri, "Modeling and Verification of Cryptographic Protocols Using Coloured Petri Nets and design/CPN," *Nordic Journal of Computing*, vol. 12, pp. 201-228, June 2005.

[15] S. Pashazadeh, "Modeling and Verification of Access Rights in TakeGrant Protection Model Using Colored Petri Nets," *International Journal of Information and Network Security (IJINS)*, vol. 2, pp. 78-90, 2013.

[16] K. Jensen, *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, 2nd ed. vol. 1. Berlin, Germany: Springer, 1996.
[17] K. Jensen and L. M. Kristensen, *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Berlin, Germany: Springer, 2009.
[18] L. C. Paulson, *ML for the Working Programmer*, 2nd ed. New York, NY, USA: Cambridge University Press, 1996.
[19] F. Butler, I. Cervesato, A. D. Jaggard, and A. Scedrov, "A Formal Analysis of Some Properties of Kerberos 5 Using MSR," presented at the 15th IEEE workshop on Computer Security Foundations, Cape Breton, NS, Canada 2002.

BIOGRAPHY OF AUTHORS

Amir Keshvari Ilkhichi is M.Sc. student of Software Engineering in Islamic Azad University of Zanjan branch in Iran. He received his B.Sc. in Software Engineering from Islamic Azad University of Shabestar branch in 2005. His research interests are computer security, analysis and verification of security protocols, and model checking of concurrent systems.

Saeid Pashazadeh is Assistant Professor of Software Engineering and chair of Information Technology Department at Faculty of Electrical and Computer Engineering in University of Tabriz in Iran. He received his B.Sc. in Computer Engineering from Sharif Technical University of Iran in 1995. He obtained M.Sc. and Ph.D. in Computer Engineering from Iran University of Science and Technology in 1998 and 2010 respectively. He was Lecturer in Faculty of Electrical Engineering in Sahand University of Technology in Iran from 1999 until 2004. His main interests are modeling and formal verification of distributed systems, computer security, and wireless sensor/actor networks. He is member of IEEE and senior member of IACSIT and member of editorial board of journal of electrical engineering at University of Tabriz in Iran.