A New Coloured Petri Net Methodology for the Security Analysis of Cryptographic Protocols

Yongyuth Permpoontanalarp
Panupong Sornkhom

Logic and Security Laboratory,
Department of Computer Engineering
King Mongkut's University of Technology Thonburi,
Bangkok, Thailand
yongyuth.per@kmutt.ac.th

Topics

- Introduction
- Problem Statements
- Objectives
- Background
- Related works
- Our new methodology
- New Attacks
- Conclusions

Introduction

- Cryptographic protocols are protocols which use cryptography techniques to achieve certain tasks & to prevent malicious parties to attack
  - Authenticated key exchange
  - Secure wireless protocol
  - Secure web protocol
  - E-payment, E-banking, etc.

Problem statements

1. Difficult to analyze cryptographic protocols
   - Attacks found after implemented, eg. UMTS, Kerberos
2. Existing Petri net methods can analyze only a single session of the protocol execution
   - Single session = one protocol run

- Many sophisticated & crucial attacks involve multiple concurrent sessions of protocol execution
  - Eg. The man-in-the-middle attack, Parallel session attacks, Reflection attacks, ...

Objectives

1. To develop a new CPN methodology for cryptographic protocols
    - Multiple concurrent sessions of protocol execution
    - Systematic method to analyze attacks
2. To apply our new CPN method to two case studies : TMN and ECS1
    - Many new attacks found in both TMN and ECS1

- Our focus
    - Message replay attacks, not cryptanalysis
    - analysis of TMN

Notations

- $S \rightarrow R : M$ means that user S sends message M to user R
- $ENC_X(M)$ = public key encryption on M by X's public key
- $E_K(M)$ = shared-key encryption on M by key K
- A = initiator, B = responder, J = server
- In = attacker

Background : TMN

- TMN = key exchange protocol for mobile communication system
  - Initiator A wants to exchange a session key with responder B by the help of server J

1. $A \rightarrow J : (B, ENC_J(K_{aj})), A$
2. $J \rightarrow B : A$
3. $B \rightarrow J : (A, ENC_J(K_{ab})), B$
4. $J \rightarrow A : B, E_{Kaj}(K_{ab})$

- Two keys: session key (K_{ab}) and A's secret (K_{aj})
- $E_K(M)$ = Exclusive-or, $ENC_X(M)$ = RSA

Related works: Petri nets for crypto protocols

- Two kinds of works

1. Attack detection
2. Semantics of crypto protocols
   - Eg. by Crazzolara and Winskel's

- Existing works on "attack detection" analyzes a single session of protocol execution only
  - Extended Petri nets : Tavares's group, Lee's group
  - CPN :
    - Dresp's work (two sequential sessions)
    - Al-azzoni et. al. 's work (two sequential sessions)
    - Suratose et. al. 's work (using simulation technique)

Related works: analysis of TMN

- 1 manual analysis : Simmon's attack on multiple sessions
- 7 approaches by formal methods
  - NRL & Interrogator : single session attacks
  - Inatest : Simmon's attack
  - Murφ by Mitchell et al : Simmon's attack + new multi-session attack
  - CSP/FDR by Lowe and Roscoe : new single session attack + new multi-session attack
  - CPN by Al-azzoni et. al. : a variant form of Murφ's multi-session attack
  - Model checking by Zhang et al : variant forms of FDR's attacks

Our new general methodology

- 5 steps

1. Building CPN graph model for representing users and attacker(s)
2. Generating state spaces
   - Decomposition : one setting at a time
     - A setting = initiator, responder, attacker role, secrets
   - Multi-session scheduling : one alternating execution of multiple sessions
3. Searching for attack states
   - Vulnerability events : to characterize attacks comprehensively

Our new general methodology

4. Extracting attack traces
   - Efficient method without the need for path searching
   - Embed an attack trace into a state as the protocol proceeds
5. Classifying attack traces by attack patterns
   - ECS1: 7,000 attacks are found
   - Attack pattern = the core of attack = minimal protocol messages

Our CPN method for TMN

- Assumptions of the protocol execution

1. There are three users : an initiator, a responder and a server. And all are honest
2. One attacker
3. The underlying encryption is perfect (Dolev & Yao’s)
    - General public key encryption scheme
4. Two concurrent sessions of the protocol
    - Sequential and non-sequential execution
5. Initiator and responder involve in one session only, but server involves up to two sessions

Our CPN method for TMN

- Attacker abilities

1. Eavesdrop, modify and drop messages during the transmission
2. Send a message to a user
3. Initiate a new session or take part in existing session
4. Impersonate any user
5. Perform crypto computation with reasonable power
6. Has its own storage with reasonable amount
7. Does not attack himself
8. At most one attacker who attacks a protocol step in a session

Our CPN method for TMN

- Two basic vulnerability events

1. The first : attacker learns a secret
    - K_{ab} and K_{aj}
2. The second : session key commitment by each user
    - A commits on K_{ab} or fake key : K_i or K_{aj}
    - B commits on K_{ab}

- Note : K_i = attacker's key

Our CPN method for TMN

- Combined vulnerability events = 1+2

  [KB1][KB2][KB3]

  where

  - KB1 = Keys known by attacker (excluding K_i)
  - KB2 = Key committed by initiator A
  - KB3 = Key committed by responder B

Our CPN method for TMN

- Five combined vulnerability events

1. $[K_{ab}][K_{ab}][K_{ab}]$
2. $[K_{ab},K_{aj}][K_{ab}][K_{ab}]$
3. $[K_{ab}][K_i][K_{ab}]$
4. $[K_{ab},K_{aj}][K_i][K_{ab}]$
5. $[K_{ab},K_{aj}][K_{aj}][K_{ab}]$

New attacks

- Events 3, 4 and 5 lead to the man-in-the-middle attack

$$A \xleftrightarrow[K_i]{} In \xleftrightarrow[K_{ab}]{} B$$

Our CPN method for TMN

- Configuration of protocol execution

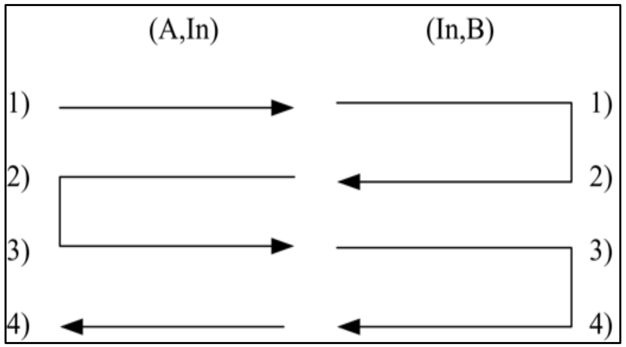
  $((S_1, S_2,\ldots,S_n), Sch, Tr)$

1. S_i is a session information (s,I,R,T,K)
   - s : session id
   - I, R, T : initiator, responder, server id
   - K : keys for each party
2. *Sch* is a multi-session schedule
   - List of session id to be executed in that order
3. *Tr* is an attack trace
   - Combined vulnerability event + a list of protocol traces

Our CPN method for TMN

- 4 employed configurations
  - *(1,A,B,J,K) & (2,In,In,J,K)*
  - *(1,A,In,J,K) & (2,In,B,J,K)*
  - *(1,In,B,J,K) & (2,A,In,J,K)*
  - *(1,In,In,J,K) & (2,A,B,J,K)*

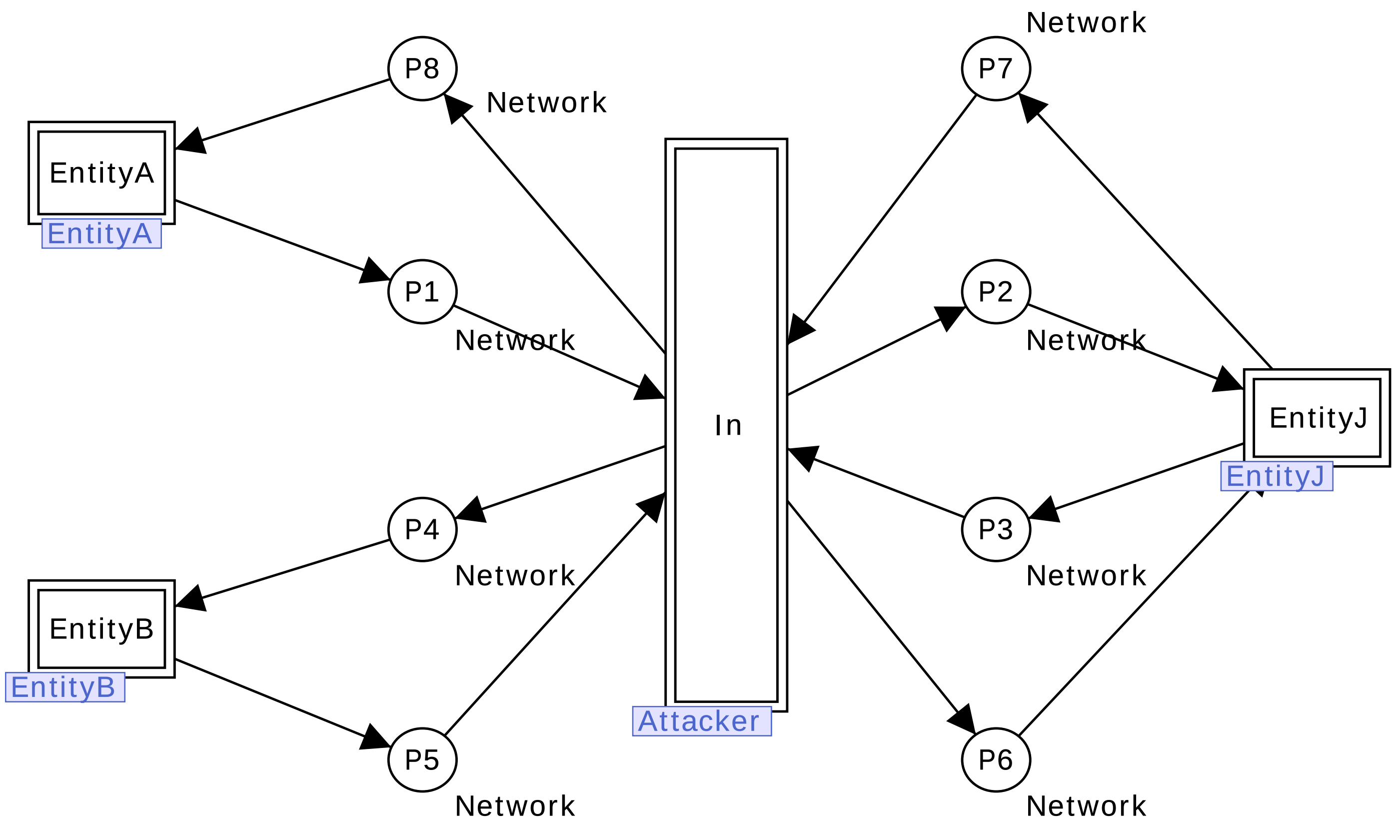
  where $K = (K_{aj}, K_{ab}, (PK_J, SK_J), K_i)$
- Sch = [1,2,2,1,1,2,2,1]



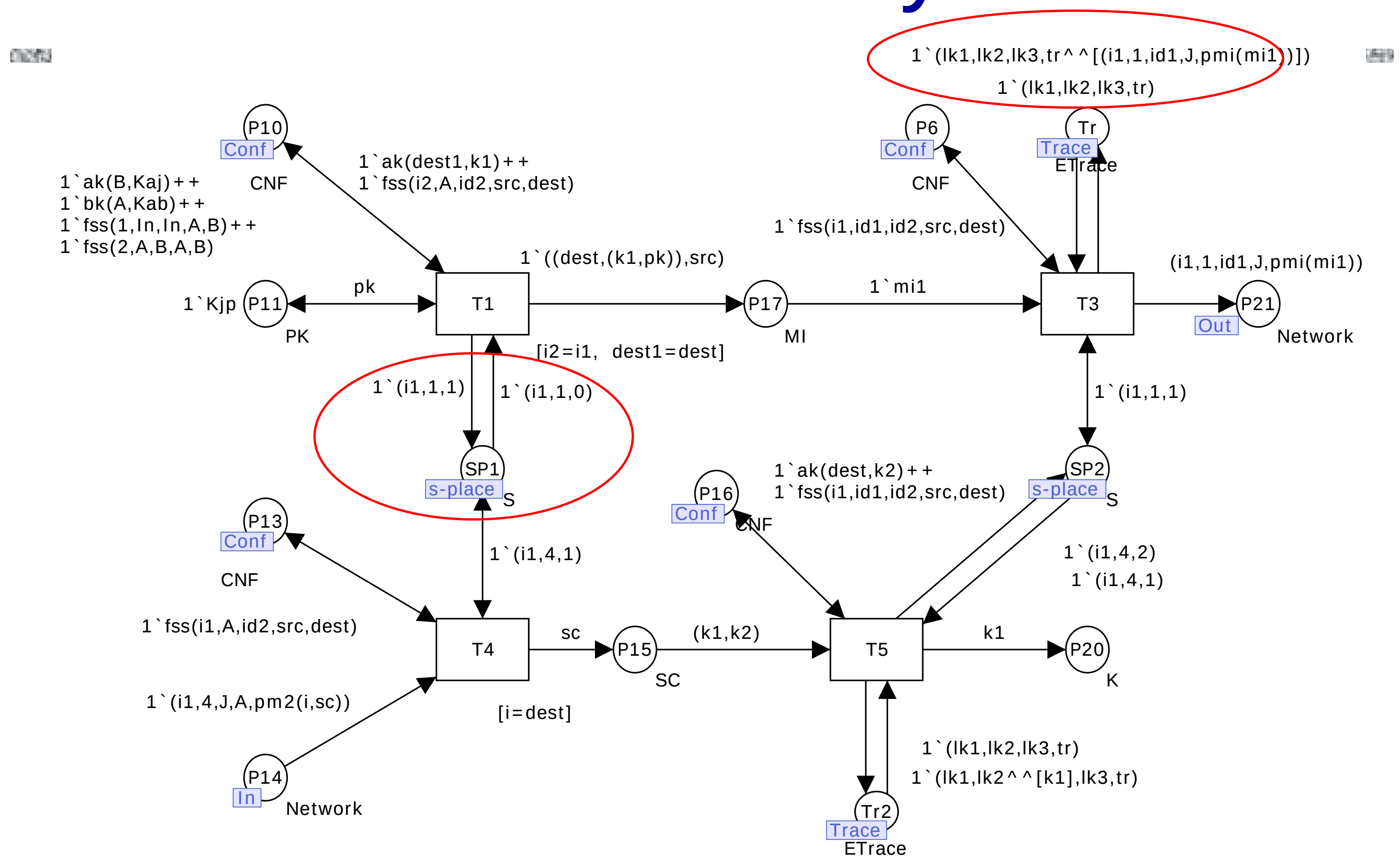
Our CPN graph model

- Based on Al-azzoni et al
- 4 levels : top, entity, sub-entity and control
- Top level = interaction between all parties
- Entity level = behaviour of each party
- Sub-entity level = specific behaviour of a party
- Control level = multi-session scheduling

Our CPN model : top level



Our CPN model : entity level



- Session state (sid,sp,st) : session id (sid), protocol step counter (sp) and states (st) where 0=ready, 1=executing, 2=finished

New attacks in TMN

- We found two new attacks which are the combined vulnerability events 4 and 5
  - 10 attack patterns for each new attack

Configurations	Event 2		Event 4		Event 5	
	Tr	Pat	Tr	Pat	Tr	Pat
1. (A,B) (In,In)	360	10	360	10	360	10
2. (A,In) (In,B)	144	4	144	4	144	4
3. (In,B) (A,In)	72	2	72	2	72	2
4. (A,B) (In,In)	36	1	36	1	0	0

Table 1: Number of Attack Traces and Patterns

New attacks : (A,B) & (In,In)

- 1st pattern for the first new attack $[K_{ab},K_{aj}][K_i][K_{ab}]$ (event 4)

1) $A \rightarrow In(J) : (B, \{K_{aj}\}PK\text{-}J), A$
   $In(J) \rightarrow J : (X2, \{K_i\}PK\text{-}J), X1$

1') $In(A) \rightarrow J : (X4, \{K_i\}PK\text{-}J), X3$

2') $J \rightarrow In(B) : X3$

2) $J \rightarrow In(B) : X1$
   $In(B) \rightarrow B : A$

3) $B \rightarrow J : (X1, \{K_{ab}\}PK\text{-}J), X2$

3') $In(B) \rightarrow J : (X3, \{K_{aj}\}PK\text{-}J), X4$

4') $J \rightarrow In(A) : X4, E_{Ki}(K_{aj})$

4) $J \rightarrow In(A) : X2, E_{Ki}(K_{ab})$
   $In(A) \rightarrow A : B, E_{Kaj}(K_i)$

where X1,X2,X3 and X4 are arbitrary identities that attacker choose

Performance

Configurations	Nodes	Arcs	Time (sec.)
1. (A,B) (In,In)	104,346	109,476	976
2. (A,In) (In,B)	73,806	77,568	523
3. (In,B) (A,In)	51,212	52,639	282
4. (A,B) (In,In)	34,160	35,095	120

Table 2. Size and Time of the generated state spaces

Conclusions

- We develop a new CPN methodology for analyzing crypto protocols
  - Multiple concurrent sessions of protocol execution
  - Decomposition & multi-session scheduling
  - Intuitive characterization of attack states by vulnerability events
  - Fast attack trace extraction
  - Attack classification by attack patterns
- Found many new attacks in TMN and ECS1