Analysis of Authentication Protocol for USB Storage Devices Using Coloured Petri Nets

Suratose Tritilanunt

Abstract—**This paper proposes an analysis of two-factor authentication protocol for a USB storage device used in digital forensic applications. The authentication protocol used in the verification is implemented by using two factor authentication technique in order to strengthen a protection of sensitive digital evidence stored in a USB storage device. By using Coloured Petri Nets as a formal tool for verifying a security of this protocol, the result is able to confirm that a proposed authentication technique satisfies security properties and does not susceptible to principle attacks.**

Index Terms—**Coloured Petri nets, CPN tool, secure authentication protocol, one time password, USB storage devices.**

I. Introduction

There are many researches concern on the lack of protection mechanism on data storage. This is because the external storage using a universal serial bus (USB) communication becomes the most popular use for storing digital data. This is because some key characteristics such as portability, large capacity, and fast data transfer rate of this peripheral devices. Because of this reason, many developers try to develop tools such as digital evidence acquisition installed into USB storage devices and use them to gather live evidence from suspect computers.

Using USB storage devices as a digital evidence acquisition brings some concern to computer security researchers. This is because there is no effective authentication mechanism on that device [1], so anyone who can run this tool is able to gather digital data that might contain sensitive information of users, such as username and password stored in that computer. Even though some developers integrate password-based authentication to USB devices, this password could be cracked and then the software installed in that device will be copied and transferred to other devices and used it illegally.

In this paper, we propose an authentication protocol based on the concept of two-factor authentication. By using username and password followed by the HMAC-based [2] one-time password (HOTP) as a second step of authentication, our proposed protocol can be integrated into the usage of USB storage devices in order to strengthen up a level of security of such devices. Moreover, the proposed protocol is able to prevent software distribution to other devices without permission from an administrator/owner because the technique and parameters that we use to generate HOTP are bind to specific USB storage device similarly to the technique of hardware dongle which is used in many software licenses nowadays. By using this technique, an additional option provided by our proposed is that the software and device will be useless when the specified period passes. This feature can prevent investigators not only using this device to gather user's information beyond their assigned jobs illegally, but also distributing software installed in this device to the others without permission or initial configuration from an owner of the software/data.

The main objectives of this paper are:

1) To design and develop a secure authentication protocol for USB storage peripherals that aim to be used in a digital forensic evidence acquisition; and 2) To validate functions of a proposed protocol, and verify security properties by using a formal method tool named Coloured Petri Nets (CPN) [3], [4].

The structure of this paper is as follow: Section II begins with the review of authentication protocols related to our proposed technique, and follows by a formal verification technique using CPN tools; Section III introduces the design of our authentication protocol followed by the experiment, CPN model, and result in Section IV; and Section V concludes our work and show some guidelines for further development.

II. Procedure for Paper Submission

This section reviews related literature on the authentication protocols, especially the ones that are designed for USB storage devices. The function, as well as the benefits and weaknesses of each technique are discussed in this section. Another review is on the use of a formal verification technique which is a Coloured Petri nets. Many researchers successfully use this formal method for analyzing and verifying the security of authentication protocols. One example tool developed for using as a Coloured Petri nets verification method is a CPN tools [4].

A. Authentication Protocols for USB Storage Devices

Using authentication protocols as an assistant technique in USB storage devices for authenticating users has been introduced by many researches. The first related literature is taxonomy on the usage of cryptographic authentication protocol in USB storage devices. This paper was published by K. Lee *et al*. [5]. In this work, they explore the authentication protocols on the USB storage devices and analyze vulnerabilities on them. This work classified vulnerabilities of USB storage devices into 12 categories based on the authentication protocols. They also grouped security on

Manuscript received July 20, 2015; revised January 15, 2016.

Suratose Tritilanunt is with the Department of Computer Engineering, Faculty of Engineering, Mahidol University, Thailand (e-mail: suratose.tri@mahidol.ac.th).

doi: 10.18178/jacn.2016.4.1.204

commercial USB products into 3 types; software-only approach, hardware-supported partitioning approach, and hardware-based encryption approach [6]. From there analysis and evaluation in all 3 types using criteria and vulnerabilities introduced in [1], [7], this paper found that almost every USB storage devices available on the today market mainly focused in the security of stored data, but not focus in a protection of the device usage and user authentication mechanism. From the judgment of the author, most of external USB storage devices are susceptible to be attacked at the authentication process.

The promising approach which mainly focuses in the design of authentication protocol to use in USB storage devices for a digital evidence acquisition application was proposed by Tritilanunt *et al*. [8]. The protocol design kept number of exchange messages in communication between a user and server as small as possible. That is because a user can use a smartphone as a device to communicate and generate a hash-based one time password (HOTP) at a digital crime scene. However, that protocol has a major vulnerability in which a dishonest user is able to compute HMAC-based one time password (HOTP) after he receives message 2 without replying message 3 to a server. That means a dishonest user does not need to authenticate himself by providing a valid username and password, as well as Diffie-Hellman key parameters [9] to a server. This is because parameters that are used as a generator for calculating HMAC-based one time password (HOTP) have been already sent by a server in message 2 (shown as an *italic text* in Fig. 1). More details of analysis and model simulation are explained in the Section IV; modeling analysis and experimental result.

1. U_A -> SV: ID_A || Job
2. SV -> U_A: Cert_{SV}||
   $\mathrm{Sign}_{SV}(g_x, p, T_1, H(\mathrm{ID}_A, \mathrm{USB}, T_1))$
3. U_A -> SV: $\mathrm{Pub}_{SV}(g_y, p)$ ||
   $E_{Ks}(\mathrm{ID}_A \| H(\mathrm{Pass}) \| H(g_y, p) \| T_1+1)$
4. U_A ->Ph: $H(\mathrm{ID}_A, \mathrm{USB}, T_1)$
5. Ph -> U_A: HOTP generation

Fig. 1. Tritilanunt's HOTP authentication protocol [8].

B. A Formal Verification Technique Using CPN Tools

The goals of security or cryptographic protocols are to provide various security services to the protocol entities. In order to provide assurance to users, the protocol developer requires formal method analysis to support the design. One famous formal method based on a simulation approach is Coloured Petri Nets (CPNs) [3]. An example of an automated tool using a concept of CPNs is CPN Tools [4], which is used to create a model as shown in Fig. 2.

Over many years, cryptographic and security protocols have been modeled and verified using Coloured Petri Nets [8], [10], [11]. The simulation of a CPN can be seen as an occurrence sequence consisting of markings that are reached and steps. To achieve the formal proof of the security protocol proposed in this paper, not only the simulation techniques can be used to explore the vulnerabilities of the protocol, but the state space analysis can be used to check the security and correctness of the protocol specification as well.

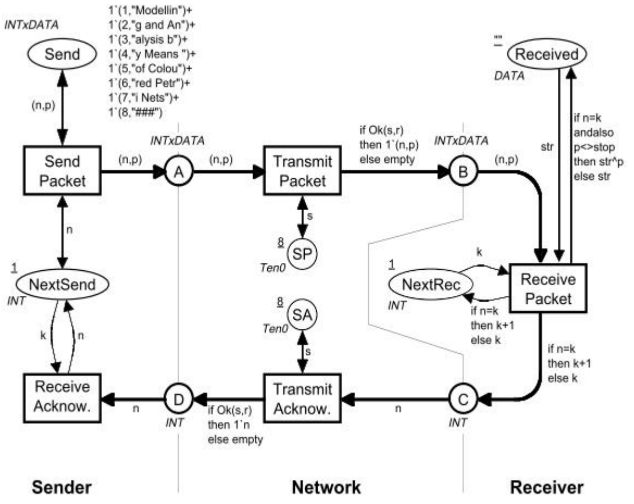

Fig. 2. Example of CPN model.

III. Two-Factor Authentication Protocol for Digital Forensic Acquisition Devices

To achieve all defined goals, we designed a secure authentication protocol using a two-factor authentication technique. Before describing more details, the protocol abbreviation and acronyms, as well as the overview of how the protocol works are explained below.

A. Abbreviations and Acronyms

Following are symbols and parameters used in the paper:

1) Parties

- SV: Server or Administrator
- U_X**:** User *X*
- Ph: Smart Phone for generating HMAC-based One Time Password (HOTP)

2) Messages

- ID_X: Identity Information of User *X*
- Cert_X: Certification of User *X*
- Sign_X(.): Signature signed by a private key from User *X* over the message (.)
- Pub_X(.): Asymmetrical encryption using a public key of User *X* over the message (.)
- *H***(.)**: Hash or Message digest of (.)
- K_s: Session key generated by Diffie-Hellman algorithm
- E_{Ks}(.): Symmetrical encryption using session key K_s over (.)
- HOTP: HMAC-based One Time Password
- Pass: Password
- USB: Unique parameter of Universal Serial Bus storage devices
- T_1: Time at T_1
- Job: Job description assigned with number which show how many times users are allowed to access a device

B. Protocol Overview

Assumption:

1) Registration phase is not required;
2) Server assigns task, USB device, and time to user
3) Server's secret parameter is a user-, session-, job-, and device-dependent specific parameter.

Before using a device to gather digital evidence, users/investigators are required to authenticate identity to a server in order to obtain a hash-based one-time password (HOTP) for use as a second parameter for authentication. The protocol consists of 6 steps including:

1) U_A -> SV: ID_A || Job
2) SV -> U_A: Cert$_{sv}$|| Sign$_{sv}$(g_x, p, T_1, H (ID_A, USB, T_1))
3) U_A -> SV: Pub$_{sv}$(g_y, p) || E_{Ks}(ID_A || H(Pass) || H(g_y, p) || T_1)
4) SV -> U_A: E_{Ks}(*Y*/*N* || *Server Secret*)
5) U_A ->Ph: H(ID_A, USB, T_1, *Server Secret*)
6) Ph-> U_A: HOTP generation.

C. Protocol Details

1) U_A -> SV: ID_A || Job

A user sends a job request to a server for accessing resources or to obtain a USB storage device. This message consists of a user identity (ID_A) concatenated with a job description (Job).

2) SV -> U_A: Cert$_{SV}$||Sign$_{SV}$ (g_x, p, T_1, H (ID_A, USB, T_1))

In the beginning of step 2, a server opens a session with a user and responds to this request by selecting Diffie-Hellman key exchange parameters g_x under modulation of p, where g is a primitive root under prime number p, and x is a random number chosen under (p-1), setting a timestamp (T_1), and computing a hash of ID_A, USB, and T_1. Finally, a server signs this message by using a private key, and then returns this signed message along with a server's certification to a user.

3) U_A -> SV: Pub$_{SV}$(g_y, p) || E_{Ks}(ID_A || H(Pass) || H(g_y, p) || T_1)

In a third step, a user obtains a server's public key by extracting it from a server's digital certification. This step provides the ability for a user to verify a server's identity via a trust certification. Then, a user calculates a session key K_s used in the next consecutive communication with a server to obtain more services (if required). By using the Diffie-Hellman key exchange protocol, a user selects y randomly under (p-1), and returns g_y and p to a server. Moreover, a user submits an identity along with a password in the hash form, and concatenates them with Diffie-Hellman key exchange parameters and a timestamp T_i. The encrypted message is returned to a server along with another asymmetrical encrypted part using a server's public key over a g_y and p parameter.

To complete the third step, a server decrypts the first part; (Pub$_{SV}$(g_y, p)), by using its private key to obtain a Diffie-Hellman parameter used for a session key K_s generation. Then, a server is able to extract the second part; (E_{Ks}(ID_A || H(Pass) || H(g_y, p) || T_1)), by using key K_s as a decryption key. As a result, a server is able to verify a user from his identity and password. If this value is valid, the server computes a *Server's secret* parameter and returns it to a client.

4) SV -> UA: E_{Ks}(Y/N || Server Secret)

If a verification process in step 3 is successful, a server will notify a user to proceed to a next step. This message also includes a *Server Secret* that is required by a user. A message is encrypted with a session key *Ks* in order to prevent from stolen by any unauthorized users.

5) UA ->Ph: H(IDA, USB, T1, Server Secret)

A user decrypts a message 4 for retrieving a server secret parameter. This value will be inserted to H (ID_A, USB, T_1)) received in step 2 for hash-based one-time password (HOTP) creation. This HOTP will be used as a 2-step authentication parameter on a USB storage device.

6) Ph-> UA: HOTP generation

Once a smart phone device receives a request from a user, a device calculates hash-based one-time password by using unique parameters as well as a *Server Secret* receive from a previous step. Fig. 3 shows the overview of the proposed protocol.

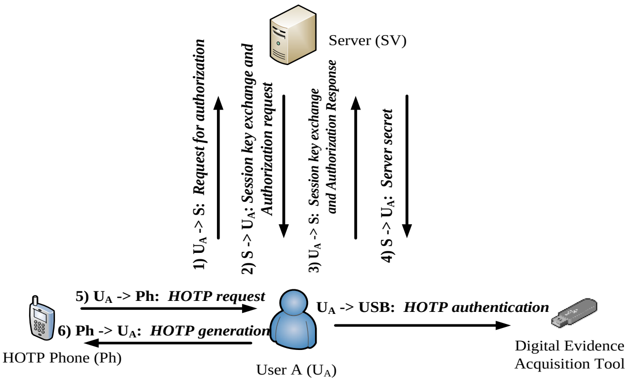
Server (SV)
1) U_A -> S: Request for authorization
2) S -> U_A: Session key exchange and Authorization request
3) U_A -> S: Session key exchange and Authorization Response
4) S -> U_A: Server secret
5) U_A -> Ph: HOTP request
6) Ph -> U_A: HOTP generation
U_A -> USB: HOTP authentication
HOTP Phone (Ph)
User A (U_A)
Digital Evidence Acquisition Tool

Fig. 3. Overview of two-factor authentication protocol.

IV. Modeling Analysis and Experimental Result

In this section, we propose a model of protocol using a CPN Tools. Our formal model is developed based on Coloured Petri Nets (CPNs) and analyzed the security by using a simulation approach provided in CPN Tools. More specific to the modeling and analysis of cryptographic protocols, some important advantages of using Coloured Petri Nets over other formal modeling and analysis techniques are the graphical representation for easy to understand and visualize the dynamic behavior of systems. Most importantly, graphical representation of Coloured Petri Nets is able to assist the protocol designer to prevent some inconsistency or ambiguous of the message sequences communicated among protocol participants in the design of protocols. Fig. 4 shows an overview of our proposed protocol which is modeled by using CPNs.

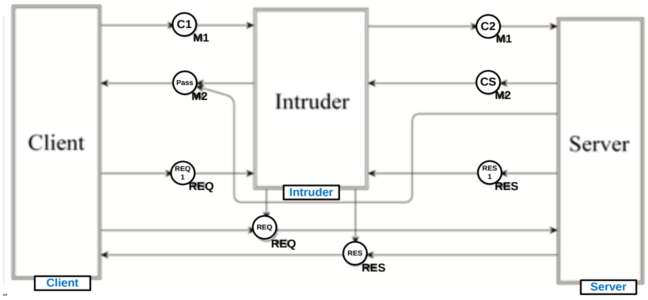

Fig. 4. Overview of our CPN model.

A. Model Declaration

For the model declaration, only important colour set (colset) are explained and shown in Table I.

TABLE I: DESCRIPTION OF COLOUR SET

colset	description
ID	user identification token
Job	job description token
USB	unique number of USB data storage devices
Timer	time token
GP	Diffie-Hellman key exchange parameters *gx*, *gy*, and *gz*
HKEY	hash of Diffie-Hellman key exchange parameters*HYP* and *HZP*
SIGN	Server's digital signature color set token
CERT	Server's certification color set token
PUBKEY	public key color set
PRI	Server's private key
HOTP	Hash-based one-time password
KEY	Session key *K_XY* and *K_YZ*
NS	Random nonce generated by server
HPASS	Password in hash-form for Client (*HPASSA*) and Attacker (*HPASSZ*)
PUBK	Diffie-Hellman key parameters encrypted by a public key
HASH1	Hash of messages
SIGNED	Signed messages using a digital signature
ENC1	Encryption of messages
MSG1-4	Messages of protocol

B. *Attacker's Ability*

For the analysis of the illegal activity, we consider two types of attacker including;

Type 1 attacker who is able to intercept messages between a client and server, attempts to generate a fake session key, and obtains parameters to computer HOTP.

Type 2 attacker (dishonest user) who does not involve the computation of message 3, but attempts to generate HOTP at step 5. This technique has been used to attack Tritilanunt *et al*. [8] protocol successfully.

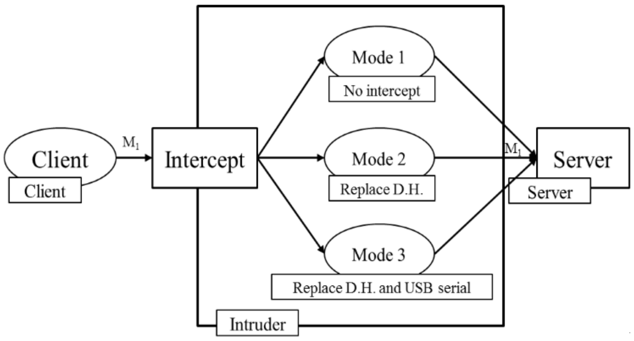

Fig. 5. Overview of attackers in CPN model.

C. *Modeling and Simulation of the Proposed Protocol*

For modeling attacker capability, there are 3 different modes as shown in Fig. 5.

Mode 1: A legitimate user honestly interacts with a server

To validate the workflow of our proposed protocol and ensure that there are no any possible chances created by attacker to cause unsafe situation, CPN Tools allows a protocol developer to check it by providing a state space analysis. The state space analysis attempts to search for all possible solution during protocol runs, and returns a number of token stored in each places in term of upper and lower bound. Since we assign 'unsafe', 'reject', and 'USB' place in the model, we can summarize that the protocol is unsafe, or a server can detect invalid packets from an attacker, or a USB device is used by a client, respectively, if there are any message tokens passed to these places. By using a state space analysis, the result is shown in Fig. 6.

```
CPN Tools state space report:
… <snip> …

Boundedness Properties
-------------------------------------------------------------------
 Best Integer Bounds
                                          Upper     Lower
    .....<snip>.....
    Main_Page'Reject1           0         0
    Main_Page'USB1              1         0
    Main_Page'Unsafe_state1  0         0

 Best Upper Multi-set Bounds
    .....<snip>.....
    Main_Page'Reject1              empty
    Main_Page'USB1                 1`USB
    Main_Page'Unsafe_state1     empty

 Best Lower Multi-set Bounds
    .....<snip>.....
    Main_Page'Reject1              empty
    Main_Page'USB1                 empty
    Main_Page'Unsafe_state1     empty
```

Fig. 6. Result from state space analysis of a legitimate user (excerpted from a full version, only related information are displayed in this figure).

- The protocol is safe because a number of token stored in an '*unsafe*' place displayed as '*Unsafe state1*' is '0' for both upper and lower bound. That means there are no any activities from users that cause messages to move into unsafe events.
- A server does not reject messages from a client because both upper and lower bound of '*reject*' place displayed as '*Reject1*' stores nothing ('0').
- A user is allowed to use a USB storage device by a server. This is because a '*USB*' place displayed as '*USB1*' contains '1' for upper bound at the beginning of protocol run, but this status is changed to '0' for lower bound at the end.

Mode 2: attacker modifies Diffie-Hellman parameters for generating a fake session key

From the simulation of CPN modeling shown in Fig. 7, a result from simulation using state space analysis shows that a place named '*reject*' contain a token as a upper bound, while a place named *'unsafe state1'* holds nothing after CPN tools finishes running all possible simulation. Moreover, a place named '*USB*' does not release a token from a server to a user. That is because a hash of password provided by an attacker is not valid, so a server rejects this message and does not permit an attacker to proceed to the next steps. By analyzing this result, we can confirm that an attacker in this scenario is unable to complete protocol steps unless he can find/guess a correct password of a legitimate user.

Mode 3: attacker/dishonest user does not send message 3 to a server, but tries to obtain HOTP from the next step.

In this simulation, a dishonest user tries to bypass authentication phase by skipping message 3. A dishonest user obtains information from message 2 and forwards it to a phone for HOTP generation. This vulnerability has been found in Tritilanunt *et al*. protocol [8]. By modeling it in CPN Tools, an experimental result from state space analysis shows that there is a token stored in an '*Unsafe state1*' because a dishonest user is able to provide all initial parameters used for computing HOTP. By considering ‘*reject*’ place, this scenario causes ‘*0*’ for both upper and lower bound because an attacker/dishonest user does not return message 3 to a server for authentication. However, a token appears at an ‘*Unsafe state1*’ place instead. This is because type 2 attacker and dishonest user are unable to submit one important parameter which is a *server secret* to a HOTP generation device. As a result, a token displayed as *1`"User bypass the procedure"* will appear at this place. Due to this reason, a place named ‘*USB*’ always holds a token and a server does not release USB device to this kind of user. The result from state space analysis is shown in Fig. 8.

```
CPN Tools state space report:
… <snip> …

Boundedness Properties
------------------------------------------------------------------
 Best Integer Bounds
                                        Upper     Lower
   .....<snip>.....
   Main_Page'Reject1           1         0
   Main_Page'USB1              1         1
   Main_Page'Unsafe_state1  0         0

 Best Upper Multi-set Bounds
   .....<snip>.....
   Main_Page'Reject1              1`"Invalid Password"
   Main_Page'USB1                 1`USB
   Main_Page'Unsafe_state1      empty

 Best Lower Multi-set Bounds
   .....<snip>.....
   Main_Page'Reject1              empty
   Main_Page'USB1                 1`USB
   Main_Page'Unsafe_state1      empty
```

Fig. 7. Result from state space analysis of Type 1 attacker (excerpted from a full version, only related information are displayed in this figure).

```
CPN Tools state space report:
… <snip> …

Boundedness Properties
------------------------------------------------------------------
 Best Integer Bounds
                                        Upper     Lower
   .....<snip>.....
   Main_Page'Reject1           0         0
   Main_Page'USB1              1         1
   Main_Page'Unsafe_state1  1         0

 Best Upper Multi-set Bounds
   .....<snip>.....
   Main_Page'Reject1              empty
   Main_Page'USB1                 1`USB
   Main_Page'Unsafe_state1      1`"User bypass the
                                  procedure"

 Best Lower Multi-set Bounds
   .....<snip>.....
   Main_Page'Reject1              empty
   Main_Page'USB1                 1`USB
   Main_Page'Unsafe_state1      empty
```

Fig. 8. Result from state space analysis of Type 2 attacker (excerpted from a full version, only related information are displayed in this figure).

D. Result from State Space Analysis of New Proposed Protocol

To solve that problem, a new proposed protocol shown in Section III.B inserts a *server secret* parameter in step 4 to guarantee that a user needs to authenticate himself in order to get this parameter from a server. This *server secret* parameter is required in the HOTP generation step. After correcting this problem, we model and simulate a new protocol in CPN Tools. The simulation using state space analysis shows that a token in ‘*unsafe state1*’ and ‘*reject*’ place are disappeared. From this point of view, a proposed authentication protocol is safe from this kind of dishonest users and attackers. This simulation using state space analysis is shown in Fig. 9.

```
CPN Tools state space report:
… <snip> …

Boundedness Properties
------------------------------------------------------------------
 Best Integer Bounds
                                        Upper     Lower
   .....<snip>.....
   Main_Page'Reject1           0         0
   Main_Page'USB1              1         0
   Main_Page'Unsafe_state1   0         0

 Best Upper Multi-set Bounds
   .....<snip>.....
   Main_Page'Reject1              empty
   Main_Page'USB1                 1`USB
   Main_Page'Unsafe_state1       empty

Best Lower Multi-set Bounds
   .....<snip>.....
   Main_Page'Reject1              empty
   Main_Page'USB1                 empty
   Main_Page'Unsafe_state1       empty
```

Fig. 9. Result from state space analysis of our proposed protocol (excerpted from a full version, only related information are displayed in this figure).

V. Discussion and Conclusion

A protocol analysis is presented in order to ensure that our proposed protocol provides not only basic security properties, but achieves our defined goals as well.

- Confidentiality

Confidentiality ensures that no unauthorized persons are able to read messages between a sender and intended recipient. As a major goal of our protocol is to have lightweight computation, the protocol design limits the usage of message encryption within the parts that contain important information only. From this exception, messages between a user and server, including password, session key generation parameters, and exchanged messages between a user and server after a user obtains HOTP, are defined as important data that need to be protected.

- Integrity

Integrity ensures that the exchanged messages between two parties are protected from modification by unauthorized parties. If modification occurs during transmission, parties should have the ability to detect these changes and reject these messages. With regards to integrity, our protocol applies both hash function and digital signature to important data. By using these techniques, a recipient ensures that exchanged messages during the communication between two entities are not modified without detection ability.

- Authentication

This security property ensures the user identity and originator of messages. As our protocol is developed for using as an authentication protocol, this is a primary concern of our protocol that must support it. By using two factor authentication including a username and password, as well as one time password, the protocol is able to support authentication services to all parties using this system.

- Replay attack
  - Replay a session key K_S: A server and user append timestamp along with g^x, g^y, and p, used in a session key generation. If a session key K_S is replayed, both parties are able to easily detect and reject it from a system.
  - Replay a User ID and password: These are appended with a timestamp. A server and user can easily notice a replay attack of this parameter. Moreover, this is encrypted by a session key K_S which is a timestamp embedded value as explained earlier.
  - Replay *HOTP*: It is useless to apply old HOTP. This is because an HOTP is a user-, USB-, and time-dependent parameter. Moreover, the technique used to generate this value is a one way hash function. As a result, replaying an HOTP is impractical for unauthorized users.
- Prevent unauthorized users from illegal usage

This is a function of HMAC-based one-time password (HOTP) that we developed. The generation of HOTP limits ability either to use USB storage devices illegally or access to resources/data stored inside a device. The generation of hash contains unique parameters of each USB device, user identity, and timestamp that are all generated and assigned by a server. The output is then embedded inside a code running in the hidden/read only partition area of USB storage devices. The next section will illustrate an overview of the computation of HOTP.

In conclusion, the main objective of this paper is to propose two-factor authentication protocol for USB storage devices. The design goal is to keep less number of exchanged messages between a user and server, but still be able to securely authenticate a legitimate user and prevent him to illegally use digital forensic acquisition tools and data stored inside that device to acquire sensitive data of other users. To verify the security of the proposed protocol, we use CPN modeling as a formal method to simulate and analyze a protocol. From that simulation and analysis using a state space technique, the proposed protocol satisfies all fundamental security, with an additional requirement to prevent unauthorized users from illegal usage on USB storage devices. We hope that the new construction as well as a proposed technique is able to help the computer security community to develop effective protocols in the future.

References

[1] H. Jeong, “Vulnerability analysis of secure USB flash drives,” *Journal of the KIISC*, vol. 17, no. 6, pp. 99-118, Dec. 2007.

[2] M. Bellare, R. Canetti, and H. Krawczyk, “Keying hash functions for message authentication,” in *Proc. the 16th Annual International Cryptology Conference on Advances in Cryptology*, 1996, pp. 1-15.

[3] K. Jensen, *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use Volume 1*, Springer-Verlag Berlin and Heidelberg GmbH & Co. K, 1997.

[4] *CPN Tools: Computer Tool for Coloured Petri Nets*, The Department of Computer Science, University of Aarhus, Denmark.

[5] K. Lee, K. Yim, and E. H. Spafford, “Reverse-safe authentication protocol for secure USB memories,” *Journal of Security and Communication Networks*, vol. 5, no. 8, pp. 834-845, 2012.

[6] K. Pin, “Attacks on and countermeasures for USB hardware token devices,” in *Proc. the 5th Nordic Workshop on Secure IT Systems Encouraging Co-operation*, Oct. 2000, pp. 35-57.

[7] L. O’Gorman, “Comparing passwords, tokens, and biometrics for user authentication,” *Proceedings of IEEE*, vol. 91, no. 12, Dec. 2003.

[8] S. Tritilanunt, “A secure authentication protocol using HOTP on USB storage devices,” in *Proc. International Conference on Information Science, Electronics and Electrical Engineering*, Apr. 2014, vol. 3, pp. 1908-1912.

[9] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.

[10] W. Dresp, “Security analysis of the secure authentication protocol by means of Coloured Petri nets,” *Communications and Multimedia Security*, 2005.

[11] A. V. Ratzer, L. Wells, H. M. Lassen *et al*., “CPN tools for editing, simulating, and analyzing Coloured Petri nets,” in *Proc. the 24th International Conference on Applications and Theory of Petri Nets (ICATPN 2003)*, 2003, vol. 2679, pp. 450-462.

Suratose Tritilanunt received a Ph.D. degree in information technology from the Information Security Institute (ISI), Queensland University of Technology, Australia in 2008. He is currently working at the Department of Computer Engineering, Faculty of engineering, Mahidol University, Thailand. He is an assistant professor working in the area of cryptography, computer and network security, vulnerability and penetration testing, and digital forensics. He publishes several research papers in computer network security and digital forensic areas. He also attended several training programs in topics about computer and network security, and digital forensics. He holds certificates such as SANS GIAC Forensic Analyst (GCFA), SANS GIAC Forensic Examiner (GCFE), SANS GIAC Penetration Tester (GPEN), Ec-council Security Analyst (ECSA), Ec-council Certified Ethical Hacker (C|EH).