ISSN 2313-1527 (PRINT)
ISSN 2313-1535 (ONLINE)

IJEIE

International Journal of Electronics and Information Engineering

Vol. 2, No. 1 (Mar. 2015)

INTERNATIONAL JOURNAL OF ELECTRONICS & INFORMATION ENGINEERING

PUBLISHING OFFICE

Min-Shiang Hwang
Department of Computer Science & Information Engineering, Asia University, Taichung 41354, Taiwan, R.O.C.
Email: mshwang@asia.edu.tw

International Journal of Electronics and Information Engineering is published both in traditional paper form (ISSN 2313-1527) and in Internet (ISSN 2313-1535) at http://ijeie.jalaxy.com.tw

PUBLISHER: Candy C. H. Lin

23-75, P.O. Box, Taichung, Taiwan 40199, R.O.C.

International Journal of Electronics and Information Engineering

Vol. 2, No. 1 (Mar. 2015)

On the Security Analysis of Protocols using Action Language

Suvansh Lal and Manik Lal Das
(Corresponding author: Manik Lal Das)

Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India
(Email: {suvansh_lal, maniklal_das}@daiict.ac.in)
(Received Sept. 13, 2014; revised and accepted Nov. 15, 2014)

Abstract

Formal security analysis is an integral part of any newly proposed security protocols. In this paper, we illustrate the formal security analysis of a protocol using Action Language (AL). The formal analysis of the protocol shows some important observations of the protocol's security claim. We provide a countermeasure to mitigate the flaws. It is shown that security protocol analysis using Action Language is an useful approach, and we believe that the work presented in the paper would encourage others to perform a formal analysis of similar protocols using ALSP.

Keywords: Action Language; Bluetooth; Information Theoretic Approach; Wired Equivalent Privacy

1 Introduction

Over the years many security protocols have been proposed for securing real-world applications with properties such as authentication, integrity, key establishment and data confidentiality. Importantly, the requirement of security varies from application to application. Therefore, designing security protocols is an interesting research problem because most of the real-world applications are supported by one or more security protocols as per applications' security requirement. In particular data exchange over communication networks is supported by many security protocols in different layers. It has been seen that many security protocols, such as authentication and key-exchange protocols, are suffer from desired security goals. The reasons behind failure of a security protocol include protocol design flaw, lack of assumptions, consideration of adversarial model, implementation issues and so on. For example, the 802.11 Wired Equivalent Privacy (WEP) protocol [19, 21], used to protect wireless communications has found serious security flaws [4]. Roughly, a large proportion of the security protocols proposed in academic literature do not succeed in their stated security claims. Most of the cases, it is observed that the security claims of the protocol have not been analyzed formally, instead, some heuristic arguments have been provided in support of the security claims. Later, the weaknesses or security flaws of the protocol are being revealed by a formal analysis of the protocol.

There are two main directions for formal security analysis of a protocol – *information theoretic approach* and *model checking approach*. Information theoretic approach [10, 16] is somewhat theorem-proving techniques with a precise adversarial model. Whereas, model checking approach [6, 11, 13] relies on tools, logic and precise security goals. However, in both approaches, logic-based successive belief derivation plays a central role. Several logic based security analysis approaches [7, 15, 17] have been evolved. Security protocol analysis using Action Language [1] is one such approach.

In this paper, we present an overview of formal analysis of a security protocol [5] using Action Language [1]. The analysis observes some security flaws in the protocol [5], which was uncover by its informal analysis. We also provide some countermeasure to mitigate the flaws.

The remainder of the paper is organized as follows: Section 2 provide some preliminaries. Section 3 presents an authenticated key exchange protocol [5], which we analyze formally using the Action language . Section 4 analyzes the security of [5] using Action language and observes some several flaws. Section 5 provides the countermeasure and we conclude the paper with Section 6.

2 Background

The protocol [5] that we analyze with Action Language are proposed for Bluetooth security. In order to make the paper self content, we give a brief overview of Bluetooth security followed by Action Language.

2.1 Bluetooth Security

This section reviews the basic security of Bluetooth that the SIG specification [22] supports, followed by observations of some of the prominent limitations of the specification.

The security architecture of Bluetooth [22] is divided into three modes: (1) non-secure; (2) service-level enforced security; and (3) link-level enforced security. In non-secure mode, a Bluetooth device does not initiate any security measures. Whereas, in service-level and link-level enforced security modes and two Bluetooth devices can establish an asynchronous connectionless link. The difference between service-level and link-level enforced security is that in the latter, the Bluetooth device initiates security procedures before the channel is established. Bluetooth supports authentication and encryption par mode of configuration. Authentication is provided by a 128-bit link key. The algorithms for authentication and encryption are based on the SAFER+ [12] cipher. When the pair is formed for the first time, they generate and share a link key $K_{init} = E_{22}[PIN, \texttt{addr}_I, RAND_R]$, where $\texttt{addr}_I$ is the address of the initiator device (say, `I`), $RAND_R$ is a random number chosen by the receiver device (say, `R`), PIN is a shared secret number that the user manually enter in both devices, and E_{22} is the algorithm based on the SAFER+ cipher. Once two devices share the initial link key K_{init}, they can renew it latter and derive a new one, known as a combination link key (say, K_{IR}). The combination key is derived as follows.

Devices `I` and `R` generate random number $RANDLK_I$ and $RANDLK_R$, respectively. Each device masks the random number by XORing it with K_{init} and sends it to other device. Both devices individually hash each of the two random numbers with the Bluetooth address of the device, using the algorithm E_{21} which is also based on the SAFER+ cipher. The two hashes are then XORed to generate the combination key K_{IR} as $K_{IR} = E_{21}(RANDLK_I, \texttt{addr}_I) \oplus E_{21}(RANDLK_R, \texttt{addr}_R)$. The old link key is then discarded. The authentication process is a challenge-response mechanism. The initiator `I` sends a challenge $RAND$ to the receiver `R` and then `R` responds to `I` with an authentication token $SRES = E_1(K_{IR}, RAND, \texttt{addr}_R)$, where $E_1(\cdot)$ is the authentication algorithm based on the SAFER cipher.

It is easy to see that the basic security model of Bluetooth depends primarily on the user's PIN (or passkey). If the user's PIN gets compromised then the secret link key is derived easily from the PIN and other parameters. In 2005, Wong et al. [20] articulate how a Bluetooth PIN can be cracked by brute force search. Jakobsson and Wetzel [9] have also observed two other attacks, namely location attack and cipher attack, in addition to the PIN cracking approach.

Table 1: Notation used in SPAK

I	Bluetooth-enabled Initiator device
R	Bluetooth-enabled Receiver device
$PRF(k; < m >)$	Keyed hashed value of message m using key k
$h(\cdot)$, $h1(\cdot)$	Cryptographically secure hash functions
$[m]^l$	The most significant l bits of string m
$a\|\|b$	Concatenation of strings a and b
$a \oplus b$	Bitwise exclusive-OR of strings a and b
$X \rightarrow Y :< m >$	X sends message m to Y over a public channel

2.2 Action Language and Smodels

Action language (AL) [1] is a formal specification language. Smodels [14] is a model finder to analyze security protocols. AL is an executable specification language for representing protocols and checking for security violations they are vulnerable to [1, 2]. AL is based on logic programming with stable model semantics(LPSM) [8]. Logic Programming enables one with declarative ease to specify the actions of the different agents in a protocol. This includes both the operational behavior of a protocol, along with the possible security attacks of an intruder. All stable models for the solution set of logic programs in AL are *minimal* and *grounded* in nature [8]. Minimalism allows one to determine exactly what happened when a protocol specification was executed. It ensures that all unwanted models are not a part of our solution set. Groundedness, on the other hand ensures that everything present in the solution set has a justification behind its presence [3]. Together, minimalism and groundedness, makes AL particularly suitable for specifying key exchange protocols. Specification of a protocol in AL requires inculcation of concepts of robotic planning [2]. Security protocol can be considered as a planning problem, where agents exchange messages which are subject to attacks by intruders. Specification of a protocol using AL is viewed as a plan to achieve the goal, and the attacks also become plans in order to achieve its goal corresponding to security violations. We use the *smodels* model checker for executing an AL protocol specification, along with the goal state set as a prospective security violation. If the model checker is able to find a model for the goal state, we say that the protocol is flawed and there exists a plan to achieve the security violation corresponding to the goal.

Table 2: Basic Sort Predicates used to specify SPAK

`nonce(N)`	A random number used once
`salt(S)`	A two-character salt
`rand(R)`	Random number
`device(`$ID_x, addr_x$`)`	Device with identity ID_x and address $addr_x$
`time(T)`	Time T at a particular instance of the protocol run

A logic program is written as a set of *rules*. A rule comprises of a *head* and a *body*, separated by a [:-]. The left hand side *head* literals hold true if all the literals on the right hand side *body* are true. A syntactically correct example of a rule in logic programming would be:

$$q : -p, s.$$

Here, let P be the logic program with S being the solution set for P. The above rule could be read as, if the literals s and p belong to the solution set S then q must also belong to the solution set S. Here, the rule is a constraint on the solution set S. Negation can also be accommodated into logic programming as *Negation as Failure* (NAF). Negated literals are accommodated into the body of clauses. The corresponding rule could be written as:

$$q\text{:- } \mathtt{not}p, s.$$

The new constraint that the given rule will imply on the solution set S could be read as, if s belongs to the set S and p is not in S then q must belong to S. In this rule , $\mathtt{not}p$ holds true, if all attempts to prove p holding true failed. We also use *lparse* [18] as a suitable front end tool to the *smodels* system to generate a grounded logic program from the specification, and this grounded logic program is executed in *smodels* to find stable models corresponding to security violations.

3 Authenticated Key Exchange Protocol for Bluetooth Security

In 2008, Das and Mukkamala [5] proposed an authenticated key exchange protocol, Simply Passkey-Authenticated Key Exchange (SPAK), for Bluetooth security. The protocol uses the user's passkey and keyed hash function for establishing the shared key. As we analyze this protocol using Action language in Section 4, we briefly review the protocol with the notations given in Table 1.

3.1 Simply Passkey-Authenticated Key Exchange (SPAK)

`Association-Step`: This step is executed when two devices want to communicate for the first time or renewing the verifier. The user enters his/her passkey pw (or PIN) and a two-character salt s into `I` and `R` manually. Both `I` and `R` compute $v = \mathrm{PRF}(pw; <\mathtt{addr_I} \| \mathtt{addr_R} \| s>)$ and store it in their database. Here, the salt is used to avoid the dictionary attack of stolen verifier. Once the association of devices is formed, the SPAK works as follows:

- s1) $\mathtt{I} \rightarrow \mathtt{R}$: $<\mathtt{addr_I}, c_1, n>$ :: `I` chooses two random numbers n and r_1, computes $c_1 = \mathrm{PRF}((v \oplus r_1); <\mathtt{addr_I} \| n>)$, where n is acted as a nonce to safeguard the protocol against replay attacks. `I` sends $(\mathtt{addr_I}, c_1, n)$ to `R`.
- s2) $\mathtt{R} \rightarrow \mathtt{I}$: $<\mathtt{addr_R}, c_2, h_1>$ :: `R` first validates n. If n is fresh (e.g., value of n is greater than the previously received n), computes $c_2 = \mathrm{PRF}((v \oplus r_2); <\mathtt{addr_I} \| \mathtt{addr_R}>)$ and $h_1 = v \oplus (r_2 \| n)$, where r_2 is a random number chosen by `R`; else abort. Then `R` sends $(\mathtt{addr_R}, c_2, h_1)$ to `I`.
- s3) $\mathtt{I} \rightarrow \mathtt{R}$: $< h_2 >$ :: `I` first extracts r_2 from h_1 as $(r_2 \| n) = h_1 \oplus v$, and then checks whether $c_2 = \mathrm{PRF}((v \oplus r_2); <\mathtt{addr_I} \| \mathtt{addr_R}>)$. If it does hold then `R` is authenticated; else abort. `I` computes $h_2 = r_1 \oplus r_2$ and sends h_2 to `R`.
- s4) `R` extracts $r_1 = h_2 \oplus r_2$ and checks whether $c_1 = \mathrm{PRF}((v \oplus r_1); <\mathtt{addr_I} \| n>)$. If it holds then `I` is authenticated; else abort.

Data Confidentiality. Once both `I` and `R` get authenticated, they establish a session key as $sk = h(r_1 \| r_2 \| n)$ through which they can exchange data encrypted under the key sk.

Table 3: Constructs used in SPAK specification

`prf`(M_1, M_2)	Psuedo-random value of M_1 and M_2.
`concat`(M_1, M_2)	Concatenation of messages M_1 and M_2 (i.e., $M_1 \| M_2$)
`xor`(M_1, M_2)	Exclusive-or of M_1 and M_2 (i.e., $M_1 \oplus M_2$)

Table 4: Predicate names and their functions

`part(M, M1)`	M as a submessage of $M1$
`symVerifier(V, A, B)`	A verifier V shared between agents A and B
`knows(A, M, T)`	Agent A knows message M at time T
`synth(A, M, T)`	Agent A synthesizes message M at time T
`says(A, B, M, T)`	Agent A's attempt of sending the message M to agent B at time T
`gets(A, M, T)`	Agent A's receipt of message M at time T
`said(A, M, T)`	Agent A' sends message M to agent B at time $T_1 < T$
`got(A, M, T)`	Agent A's receipt of message M at time $T_1 < T$

4 Formal Analysis of SPAK using ALSP

4.1 AL Specification of SPAK

We specify the protocol by considering the basic sort predicates, to characterize the basic components of the SPAK protocol. These components include agents, nonce, time and other entities which form an integral part of a security protocol. We state clearly the background theory(initial state of the protocol) which contains rules describing, how a message is composed, understood, manipulated, encrypted and decrypted by the agents. It also includes the properties of shared keys and how information is attained by agents participating in a protocol. The basic sort predicates used in the AL specification of the SPAK protocol are shown in Table 2.

A special sort predicate `msg(M)` is also defined, which means that M is a valid message appearing in a protocol run. Then, we specify a few basic constructors that symbolize cryptographic operations, concatenations and hashing of messages as required by the protocol. Table 3 represents a few classical constructs used in the protocol specification.

We also specify predicates that define the properties of messages and keys that are used in the protocol. In addition, definition of the ability of agents to construct, send, receive and understand these messages is also an important part of our protocol specification. As suggested in [3], the predicate names in most part of our AL specification for SPAK are fairly intuitive and represent the action or property after which they are named. We give a brief mention of these predicates in Table 4.

We now initiate the protocol specification with a definition of all the messages that can be used in the protocol. Ideally any message can be used in a protocol run, so we should define them all through induction. We note that inductively defined messages would have infinitely many ground instances [1]. As SPAK is a three-step protocol, we can easily distinguish the three messages along with their sub messages, transmitted at different stages of its execution. For example, consider the first message to be transmitted in step (s1) of the SPAK protocol. The AL specification for the message and all its valid submessages could be written as:

$$msg(concat(addr_I, prf(xor(V, R1), concat(addr_I, N)), N)) : -$$
$$rand(R1), nonce(N), device(I, addr_I), symVerifier(V, I, R), passkey(P), salt(S).$$
$$msg(prf(xor(V, R1), concat(addr_I, N))) : -$$
$$rand(R1), nonce(N), device(I, addr_I), symVerifier(V, I, R), passkey(P), salt(S).$$
$$msg(N) : -nonce(N).$$
$$msg(addr_I) : -device(I, addr_I).$$
$$msg(concat(addr_I, N) : -device(I, addr_I), nonce(N)$$
$$msg(xor(V, R2) : -symVerifier(V, I, R), rand(R2), passkey(P), salt(S).$$
$$msg(R1) : -rand(R1).$$

The AL specification for a security protocol also represents the ability of agents to modify and manipulate messages

in a protocol run. To do this, message *parts* are inductively defined based on the protocol constructors.
Example:

$$part(M, M) : -msg(M).$$
$$part(M, concat(M1, M2)) : -msg(M), msg(M1), msg(M2), part(M, M1).$$
$$part(M, concat(M1, M2)) : -msg(M), msg(M1), msg(M2), part(M, M2).$$

We now describe the properties of the verifiers defined below used by the agents in the protocol. There exists a unique symmetric verifier for each pair of communicating devices. For a verifier to be generated, it is essential that there exits two Bluetooth enabled devices(`I` and `R`), a passkey (PW) and a two-character salt (S).

$$verifier(prf(P, concat(addr_I, addr_R), S), I, R) : -device(I, addr_I), device(R, addr_R), I \neq R, passkey(P), salt(S).$$
$$symVerifier(V, I, R) : -verifier(V, I, R), device(I, addr_I), device(R, addr_R).$$

Now we focus on specifying the ability of an agent to understand and synthesize these messages in a protocol run. Intuitively, the `knows` predicate is used for modelling the ability of agents to acquire information from messages they have either received or transmitted. If an agent possesses the knowledge of a message `M2` then, he/she would possess the knowledge of a message `M1` which is a valid sub-message of `M2`. This enables an agent to extract useful message parts from concatenated or xor-ed messages. To enhance the readability of our protocol specification, we skip the sort predicates in the *body* clause of our rules.

$$knows(I, M, T) : -said(I, R, M, T)$$
$$knows(I, M, T) : -got(I, M, T)$$
$$knows(I, M, T) : -knows(I, M1, T), part(M, M1)$$
$$knows(I, M, T) : -knows(I, concat(M1, M2), T), part(M, M1)$$
$$knows(I, M, T) : -knows(I, concat(M1, M2), T), part(M, M2)$$
$$knows(I, M, T) : -knows(I, xor(M1, M2), T), knows(I, M2, T), part(M, M1)$$
$$knows(I, M, T) : -knows(I, xor(M1, M2), T), knows(I, M1, T), part(M, M1).$$

A few more rules that cater specifically to the SPAK protocol are defined as follows:

$$knows(I, xor(M1, M3), T) : -knows(I, xor(M1, M2), T), knows(I, xor(M2, M3), T)$$
$$knows(I, xor(M1, M3), T) : -knows(I, xor(M2, M1), T), knows(I, xor(M2, M3), T).$$

Similarly, we specify the ability of an agent to synthesize a message in a protocol run. The rules defining the synthesis of messages ensure that an agent can construct a message if and only if it can understand or synthesize, all the valid submessages of that message.

$$synth(I, M, T) : -knows(I, M, T)$$
$$synth(I, prf(M1, M2), T) : -knows(I, M1, T), knows(I, M1, T)$$
$$synth(I, concat(M1, M2), T) : -synth(I, M1, T), synth(I, M2, T)$$
$$synth(I, xor(M1, M2), T) : -synth(I, M1, T), synth(I, M2, T).$$

We specify the actions related to transmission and receipt of messages. Most of the specification in this part is protocol independent, we refer to [1, 2, 3] for a detailed description.

$$got(R, M, T + 1) : -gets(R, M, T)$$
$$said(I, R, M, T + 1) : -says(I, R, M, T)$$
$$got(R, M, T + 1) : -got(R, M, T)$$
$$said(I, R, M, T + 1) : -said(I, R, M, T).$$

An important security requirement of an authentication protocol is session freshness. A message `msg(M)` is considered to be *used* at time `T`, if an agent says that message in a protocol run at any time $T' \leq T$. The predicate `usedPar(M, T)` holds true if two different agents use a same message `msg(M)` at a time `T`, in two parallel runs of a protocol. A message is *fresh* at time `T`, if it has not been used in any parallel or previous runs of the protocol.

$$
\begin{aligned}
&used(M, T+1) :- used(M, T)\\
&used(M, T) :- says(I, R, M1, T), part(M, M1)\\
&usedPar(M, T) :- says(I1, R1, M1, T), part(M, M1), says(I2, R2, M2, T),\\
&\quad part(M, M2), I1 \neq I2, R1 \neq R2, M1 \neq M2.\\
&fresh(M, T) :- notused(M, T), notusedPar(M, T).
\end{aligned}
$$

We specify message validation rules, which enables the agents to proceed in a protocol run, only if a message or its component has been verified. The curly brackets { } around, the *head* predicate of a rule indicates that there may exists stable models as solution sets of our logic program both with and without the existence of predicate itself in these models.

$$
\begin{aligned}
&\{says(I, R, concat(addr_I, prf(xor(V, R1), concat(addr_I, N)), N), T)\} :-\\
&\quad fresh(N, T), fresh(R1, T), symVerifier(V, I, R), I \neq R,\\
&\{says(R, I, concat(addr_R, prf(xor(V, R2), concat(addr_I, addr_R), concat((xor(V, R2), N)), T)\} :-\\
&\quad fresh(R2, T), got(R, concat(addr_I, prf(xor(V, R1), concat(addr_I, N), N), T), symVerifier(V, I, R), I \neq R\\
&\{validC2(I, R, prf(xor(V, R2), concat(addr_I, addr_R), T)\} :-\\
&\quad knows(I, addr_R, T), knows(I, R2, T), got(I, concat(addr_R, prf(xor(V, R2), concat(addr_I, addr_R),\\
&\quad concat(V, R2, N)), T), symVerifier(V, I, R)\\
&\{says(I, R, xor(R1, R2), T))\} :-\\
&\quad said(I, R, concat(addr_I, prf(xor(V, R1), concat(addr_I, N), N), T),\\
&\quad got(I, concat(addr_R, prf(xor(V, R2), concat(addr_I, addr_R), concat(V, R2, N)), T),\\
&\quad knows(I, R2, T), symVerifier(V, I, R), I \neq R\\
&\{validC2(I, R, prf(xor(V, R2), concat(addr_I, addr_R), T)\}\\
&\{validC1(R, I, prf(xor(V, R1), concat(addr_I, N)), T)\} :-\\
&\quad knows(R, addr_I, T), knows(R, R1, T), knows(R, N, T),\\
&\quad got(R, concat(addr_I, prf(xor(V, R1), concat(addr_I, N), N), T), symVerifier(V, I, R), I \neq R.
\end{aligned}
$$

Table 5: Execution of the goal state corresponding to `attack1(T)`

```
smodels version 2.26.  Reading...done
False
Duration 10.390
Number of choice points:  0
Number of wrong choices:  0
Number of atoms:  30816
Number of rules:  186638
Number of picked atoms:  0
Number of forced atoms:  0
Number of truth assignments:  5404
Size of search space (removed):  0 (0)
```

This concludes the specification of the SPAK protocol.

4.2 Attacks on SPAK

We set the goal states for our protocol run, each of which corresponds to a security violation. This incorporates the approach as suggested in [1, 2, 3] and plan attacks on SPAK so as to check the basic security properties like confidentiality and authentication. The first rule `attack1(T)` checks if there exists a goal state from a protocol initiators point of view, where a dishonest agent, say Spy, is able to achieve the key elements `(R1, R2, N)` for a session.

$$attack1(T) : -said(A, B, concat(addr_I, prf(xor(V, R1), concat(addr_I, N), N), T)),$$
$$got(A, concat(addr_R, prf(xor(V, R2), concat(addr_I, addr_R), concat(xor(V, R2, N)), T),$$
$$said(A, B, xor(R1, R2), T), knows(spy, N, T), knows(spy, R1, T), knows(spy, R2, T), A \neq spy, B \neq spy.$$

Table 5 is an execution of the specification for the goal state corresponding to `attack1(T)`:
Command: `lparse spak.lp at1.lp | smodels`

The `attack1(T)` shows that the result is False, that is, the Spy agent cannot get any useful information by this attempt.
The `attack2(T)` defined below is aimed at checking whether SPAK provides *mutual authentication* to its participating entities.

$$attack2(T) : -got(B, concat(addr_I, prf(xor(V, R1), concat(addr_I, N), N), T)$$
$$says(B, A, concat(addr_R, prf(xor(V, R2), concat(addr_I, addr_R), concat(xor(V, R2), N)), T))$$
$$got(B, xor(R1, R2), T)$$
$$notsaid(A, B, concat(addr_I, prf(xor(V, R1), concat(addr_I, N), N), T)$$
$$notsaid(A, B, xor(R1, R2), T), A \neq spy, B \neq spy.$$

Table 6 is an execution of the specification for the goal state corresponding to `attack2(T)`.
Command: `lparse spak.lp at2.lp | smodels`

Table 6: Execution of the goal state corresponding to `attack2(T)`

```
smodels version 2.26.  Reading...done
True
Duration 109.827
Number of choice points:  208
Number of wrong choices:  24
Number of atoms:  125348
Number of rules:  1040351
Number of picked atoms:  254233
Number of forced atoms:  436
Number of truth assignments:  40149094
Size of search space (removed):  752 (221)
```

The `attack2(T)` shows that the result is True, that is, it shows how an adversary can false authenticate himself to an honest agent by observing the previous sessions of the protocol. The plan for the security violation is as follows:
Run 1:

I $\rightarrow$ R: concat(addr$_I$,C1,N)
R $\rightarrow$ I: concat(addr$_R$,C2,h1)
I $\rightarrow$ R: h2

Spy Observes: C1, C2, h1, h2, N.
Initially, the Spy agent eavesdrops a protocol run and observes all the individual message segments that are transmitted between honest agents I and R. The Spy then attempts to run with `R` masquerading as `I`.
Run 2:

Spy(I) $\rightarrow$ R: concat(addr$_I$, C1'=C2, N'=addr$_R$)
R $\rightarrow$ I: concat(addr$_R$, C2', h1')

Spy Observes: C1′, C2′, h1′
Spy computes: xor(h1, h1′), knows N′, N
Therefore, Spy can compute xor(R2, R2′), and then, Spy synthesizes h2′=xor(R2, R2′).
Spy(I) $\rightarrow$ R: h2'.

Now, `R` validates $R2'$ that it generated for the current session and assumes that an honest agent `I` is participating in the protocol run; however, s/he is actually interacting with a Spy.

In order to check the forward security property in SPAK, we assume that an adversary has the verifier v known to him. We reflect this assumption in our protocol specification by allowing all the agents other than I and R to know the `verifier(V, A, B)`. Then we execute the modified specification with goal state as `attack1(T)`, and find a model that shows SPAK does not ensure forward secrecy. The plan for the security violation is:

Run 1:

I $\rightarrow$ R: concat(addr$_I$,C1,N)
R $\rightarrow$ I: concat(addr$_R$,C2,h1)
I $\rightarrow$ R: h2.

Spy Observes: C1, C2, h1, h2, N.
Spy knows: v.

Using these information the Spy is able to determine the session's secret $r1$ and $r2$ as follows:
xor(h1, v) = xor(R2, N); R2
xor(h2, R2) = R1; R1
h(concat(R1, R2, n)) = k; k(session key)
Spy knows: k, R1, R2, n.

This shows that SPAK does not provide forward secrecy.

5 Suggested Improvements and Conclusions

We have illustrated a security protocol analysis using ALSP. The analysis, presented in the paper, observes some security flaws which were uncover in [5]. We also suggested some possible countermeasure to mitigate the flaws. The attack trace suggests that steps (s1) and (s2) of SPAK have to be repaired. The step (s1) could be repaired by concatenating the address of the initiator and $C1$ with N' rather than N, where $N' = N \oplus v$. The step (s2) could be repaired by modifying $C2$ to $C2'$, where $C2'$= PRF(($v \oplus R2$), (addr$_I\|$ (addr$_R \oplus N \oplus v$))). These improvements reduce the ability of an agent to manipulate messages intercepted in steps (s1) and (s2), thereby, eliminating the possibilities of the attacks. We believe that ALSP based protocol analysis is an useful approach for formal analysis of security protocols, and the work presented in the paper would encourage others to perform formal analysis of similar security protocols using ALSP.

References

[1] L. C. Aiello and F. Massacci, "An executable specification language for planning attacks for security protocols," in *Proceedings of the IEEE Computer Security Foundation Workshop*, IEEE Computer Society, pp.88–103, 2000.
[2] L. C. Aiello and F. Massacci, "Verifying security protocols as planning in Logic Programming," *ACM Transactions on Computational Logic*, vol. 2, no. 4, pp. 542–580. 2001.
[3] L. C. Aiello and F. Massacci, "Planning attacks to security protocol: Case Studies in logic programming," in *Computational Logic: Logic Programming and Beyond*, LNCS 2407, Springer-Verlag, pp.113-169, 2002.
[4] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: the insecurity of 802.11," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pages 180–189, 2001.
[5] M. L. Das and R. Mukkamala, "Revisiting Bluetooth security," in *Proceedings of the International Conference on Information Systems Security (ICISS 2008)*, LNCS 5353, Springer, pp.132–139, 2008.
[6] B. Donovan, P. Norris, G. Lowe, "Analyzing a library of security protocols using Casper and FDR," in *Proceedings of the Workshop on Formal Methods and Security Protocols*, 1999.
[7] F. J. T. Fábrega, "Strand spaces: proving security protocols correct," *Journal of Computer Security*, vol. 7, no. 2-3, pp.191–230, 1999.
[8] M. Gelfond and V. Lifschitz, "The Stable Model Semantics for Logic Programming," in *Proceedings of the International Conference on Logic Programming*, MIT-Press, pp.1070–1080, 1988.
[9] M. Jakobsson, and S. Wetzel, "Security weaknesses in Bluetooth," in *Proceedings of the RSA Conference*, LNCS 2020, Springer–Verlag, pp.176–191 2001.
[10] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, CRC Press, 2008.
[11] G. Lowe, "Casper: A compiler for the analysis of security protocols," in *Proceedings of the IEEE Computer Security Foundations Workshop*, pp.18–30, 1997.
[12] J. Massey, G. Khachatrian, and M. Kuregian, "Nomination of SAFER+ as Candidate Algorithm for the Advanced Encryption Standard," in *Proceedings of the AES Candidate Conference*, 1998.

[13] J. C. Mitchell, V. Shmatikov, and U. Stern, "Finite-State Analysis of SSL 3.0," in *Proceedings of USENIX Security Symposium*, pp. 201–216, 1998.

[14] I. Niemela and P. Simmons, "Smodels - an implementation of Stable Model and Well-founded Semantics for Normal Logic Programs," in *Proceedings of the International Conference on Logic Programming and Nonmonotonic Reasoning*, LNAI 1265, Springer-Verlag, pp.420–429, 1997.

[15] A. D. Rubin, "Nonmonotonic cryptographic protocols," in *Proceedings of the Computer Security Foundations Workshop*, pp.100–116, 1994.

[16] C. E. Shannon, "Communication Theory of Secrecy Systems," Bell System Technical Journal, 28 (4): 656–715. 1949.

[17] D. X. Song, "Athena: a new efficient automatic checker for security protocol analysis," in *Proceedings of the IEEE Computer Security Foundations Workshop*, pp.192–202, 1999.

[18] T. Syrjanen, "Implementation of local grounding for logic programs with stable model semantics," Technical Report B18, Helsinky Univ. of Technology, 1998.

[19] J. R. Walker, "Unsafe at any key size; an analysis of the WEP encapsulation," IEEE Document 802.11-00/362, 2000.

[20] F. L. Wong, F. Stajano, and J. Clulow, "Repairing the Bluetooth pairing protocol," in *Proceedings of the International Conference on Security Protocols*, LNCS 4631, Springer-Verlag, pp.31–45, 2005.

[21] "An Overview of 802.11 Wireless Network Security Standards and Mechanisms," SANS Institute InfoSec Reading Room. http://www.sans.org/reading-room/whitepapers/wireless/overview-80211-wireless-network-security-standards-mechanisms-1530 [Accessed December 2013]

[22] "Bluetooth Special Interest Group," Bluetooth Security Specification, Specification of the Bluetooth System, 1.2, 2003.

Suvansh Lal has received his B.Tech. degree in Information and Communication Technology from DA-IICT, Gandhinagar in 2009. He then completed his master degree from Institut sup?rieur d'Electronique de Paris. Currently, he works with industry as a security architect. His research interests include Information Security.

Manik Lal Das received his Ph.D. degree from Indian Institute of Technology, Bombay. He is an Associate Professor in Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India. He is a member of IEEE and Cryptology Research Society of India. His research interests include Cryptography and Information Security.

An Attribute Based Key Agreement Protocol Resilient to KCI Attack

Majid Bayat[1], Mohammad Reza Aref[2]
(Corresponding author: Majid Bayat)

Department of Mathematical Sciences and Computer, University of Kharazmi, Tehran, Iran[1]
ISSL, Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran[2]
(Email: bayat@khu.ac.ir)
(Received Dec. 8, 2013; revised and accepted May 27, 2014)

Abstract

Attribute based key agreement protocols are kind of identity based protocols which the identities consist of descriptive attributes. Fujioka et al. in 2011 suggested an attribute based key agreement protocol resilient to ephemeral key reveal attack. In this paper, we show that the Fujioka protocol is vulnerable to key compromise impersonation (KCI) attack and consequently a secure attribute based key agreement protocol is introduced. We prove the security of our protocol in the random oracle model.

Keywords: Attribute based key agreement protocols; eCK Model; Key agreement protocols; Random oracle model

1 Introduction

Key agreement protocols enable two users to establish a shared secret key in an insecure and public channel such that the other users cannot compute the key. The key agreement protocols play an essential role in cryptographic systems and any of their weakness results a destructive attack. Hence, there are several security requirements mentioned for key agreement protocols that are listed in the following [4, 9]:

- Known Session Key Security: This property emphasises that if an adversary obtains a session key, the session keys of the coming sessions remain as secure.
- Forward Secrecy: This security notifies that by revealing the long term private keys of the two users (perfect forward secrecy) or one of the users (weak forward secrecy), the adversary cannot obtain the previous session keys.

  Strong security is a kind of forward secrecy stating if the short term private keys of the two users or one of their long term private key and short term private key of the other user is revealed, the previous session keys should not be computed by the adversary.
- Key Compromise Impersonation (KCI): Let A and B be the two users. Obviously if the adversary has the long term private key of A, it can forge A. KCI states that the adversary can not forge B by obtaining the long term private key of A.
- Unknown key security: Let A and B be the two users of a key agreement protocol. This property states that an active adversary C cannot interfere in the protocol execution such that A believes that it makes a session key with B, while B knows C as his participant in the protocol.

A kind of key agreement protocols is attribute based key agreement protocols(ABKA) that they are identity based key agreement protocols (IBKA), where identity of the users are a set of descriptive attributes. In these protocols each user can establish a session key with the other user if their attribute sets satisfy a predefined policy. Wang et al. [12, 13, 14] suggested three ABKA, which in fact, they are identity based protocols because there are not any policy in their protocols. Gorantla et al. [6] proposed an ABKA using a key encapsulation method (KEM), where an access structure is defined and two users whose attribute sets satisfy the access structure can establish a common session key. Birkett et al. [3] posed an ABKA, which each user uses an attribute based signature in the protocol. Fujioka et al. [5] and Yoneyama [15] state that the Birkett protocol is vulnerable to ephemeral key leakage and thus it does not have strong security. They use NAXOS method [9] to secure their protocols against ephemeral key leakage attack. In the Fujioka protocol, the policy is on the long term private keys and in the Yoneyama protocol the policy is on the cipher text.

In this study, we show that the Fujioka protocol is vulnerable to KCI attack and then we introduce a secure attribute based key agreement protocol which is secure in the random oracle model. We utilize the security model introduced by Fujioka [5] that is an extended model of LaMacchia et al. [9] security model for key agreement protocols. In Section 2, we bring a number of preliminaries and in Section 3, we review the Fujioka protocol and describe how it is vulnerable to KCI attack. Section 4 illustrates the proposed ABKA protocol and its security. In Section 5, a brief conclusion is discussed.

Attribute based encryption: The concept of attribute based encryption (ABE) was introduced by Sahai and Waters in Eurocrypt'05 [10]. Their scheme is a kind of identity based encryption (IBE) which the user's identity contains descriptive components called attribute. An encrypted message with identity W can be decrypted by any user whose identity contains at least d components of W that d is a threshold. This construction enables an IBE system to admit a limited error and this property is called error-tolerance. Until now there have been suggested many attribute based encryption (ABE) schemes that are categorized to the cipher text policy ABE [2, 7] and the key policy ABE [1, 8].

2 Preliminaries

Bilinear Pairing:

Let G_1 and G_2 be two cyclic groups of order of a prime number p and g be the generator of G_1. $e : G_1 \times G_1 \to G_2$ is a bilinear paring if the following conditions are hold:

- For all $X, Y \in G_1$ and $a, b \in Z_p$, $e\left(X^a, Y^b\right) = e\left(X, Y\right)^{ab}$.
- $e\left(g, g\right) \neq 1$.
- For all $X, Y \in G_1$, there is an efficient algorithm to compute $e\left(X, Y\right)$.

Bilinear Diffie Hellman Problem (BDH):

Let $e : G_1 \times G_1 \to G_2$ and g^w, g^v and g^u be the given values of G_1. The problem is to find the value $e\left(g, g\right)^{uvw}$, where $u, v, w \in Z_p$.

Decisional Bilinear Diffie Hellman Problem (DBDH):

Let $e : G_1 \times G_1 \to G_2$. The input of the problem are the values g^u, g^v, g^w and g^z of G_1, where $u, v, w, z \in Z_p$. The output is 1 if $e\left(g, g\right)^{uvw} = e\left(g, g\right)^z$ and otherwise the output is 0.

Gap Bilinear Diffie Hellman Problem (GBDH):

This problem is a BDH problem, in which the oracle of DBDH is available.

Security Model:

We use a security model similar to the security model introduced by Fujioaka et al. [5] which is an extension of the security model of LaMacchia et al. [9]. This model emphasizes on the strong security and according to the model, an ABKA protocol consists of three algorithms:

- **Key Generation:** This algorithm receives a security parameter 1^k as the input and the outputs are the master private key msk and the master public key mpk for PKG.
- **Key Extraction:** This algorithm receives the master private key, the mater public key and an attribute set δ_J of the user U_J and its output is the private key D_J.
- **Key Exchange:** Let a user U_A with attribute set δ_A and a user U_B with attribute set δ_B be two participants of the key agreement protocol with n flows. The user U_A starts the protocol and computes a message m_1 using the attribute set δ_A, the master public key and the attribute set δ_B, $Message\left(mpk, \delta_A, \delta_B, D_A\right) \to m_1$, in which $Message$ is the algorithm of the computing messages of the users. Then the message m_1 is sent

to U_B. The user U_X ($X = A$ or $X = B$) computes the message m_i after receiving the message m_{i-1}, $Message(mpk, \delta_X, \delta_{\bar{X}}, m_1, m_2, \cdots, m_{i-1}) \rightarrow m_i$, for all $i = 2, \cdots, n$. Subsequently U_X sends the message m_i to the user $U_{\bar{X}}$, that $U_{\bar{X}}$ is the partner of U_X in the protocol. After receiving the n-th message, U_X ($X = A$ or $X = B$) computes the session key as follows: $SessionKey\,(mpk, \delta_X, \delta_{\bar{X}}, D_X, m_1, m_2, \cdots m_n) \rightarrow K$. If the attribute set of U_A and U_B satisfy the policies of the PKG, then they can compute the session key K.

Session: A session is activated with the message (I, δ_A, δ_B) or $(R, \delta_B, \delta_A, m_1)$, in which I and R present the initiator and responder of the protocol, respectively. If U_A is activated with the message (I, δ_A, δ_B), then U_A is the initiator of the protocol and if U_B is activated with $(R, \delta_B, \delta_A, m_1)$, then U_B is called the responder of the protocol. The incoming messages to the initiator are in the form of $(I, \delta_A, \delta_B, m_1, \cdots, m_{k-1})$ from the responder and the initiator U_A outputs m_k. The incoming messages to the responder are in the form of $(R, \delta_B, \delta_A, m_1, \cdots, m_k)$ and the responder U_B outputs m_{k+1}. After sending or receiving the message of the n-th flow, U_A and U_B compute the session key. Session ID of the initiator is defined as follows.

$$sid = (I, \delta_A, \delta_B, m_1)\,, (I, \delta_A, \delta_B, m_1, m_2, m_3) \cdots, (I, \delta_A, \delta_B, m_1, \cdots, m_n)\,.$$

If u_B is the responder of the session, then the session ID of U_B is as follow:

$$sid = (R, \delta_B, \delta_A, m_1, m_2)\,, (R, \delta_B, \delta_A, m_1, m_2, m_3, m_4) \cdots, (R, \delta_B, \delta_A, m_1, \cdots, m_n)\,.$$

We say that a session is completed if a session key is computed in the session. The *matching session* of a completed session $(I, \delta_A, \delta_B, m_1, \cdots, m_n)$ is the completed session with identifier $(R, \delta_B, \delta_A, m_1, \cdots, m_n)$ and vice versa. **The Adversary:** Let a polynomial probabilistic adversary A controlling all communications of the protocol with the following queries:

- Send(message): The message is $(I,\ \delta_A,\ \delta_B,\ m_1,\ \cdots,\ m_{2k-1})$ or $(R, \delta_B, \delta_A, m_1, \cdots, m_{2k})$ which the simulator answers the question according to the protocol.
- SessionKeyReveal(sid): The adversary A obtains the session key of the complete session sid.
- EphemeralReveal(sid): The ephemeral secret keys of the session sid are given to the adversary.
- StaticReveal(δ): A obtains the long term private key (the output of Key Extraction algorithm) of the attribute set δ.
- MasterReveal: The adversary receives the master private key of the PKG.
- EstablishParty(U_l, δ_l): This allows the adversary to register PKG instead of the user U_l and consequently the adversary can play the role of U_l with attribute set δ_l in the protocol. After this query, U_l is called a dishonest user.

Definition of Freshness: Let $sid^* = (I,\ \delta_A,\ \delta_B,\ m_1,\ \cdots,\ m_n)$ or $sid^* = (R, \delta_B, \delta_A, m_1, \cdots, m_n)$ be a complete session between two users U_A and U_B with attribute sets δ_A and δ_B respectively. The attribute sets satisfy the predefined policy. Let the matching session of sid^* be $\overline{sid^*}$, the session sid^* is fresh if none of the following conditions are hold:

1) Let sid^* has the matching session $\overline{sid^*}$. The adversary asks SessionKeyReveal(sid^*) or SessionKeyReveal($\overline{sid^*}$).
2) If sid^* has the matching session $\overline{sid^*}$, A asks either of the following questions:
   - $StaticReveal(\delta_B)$ and $EphemeralReveal(sid^*)$.
   - $StaticReveal(\delta_A)$ and $EphemeralReveal(\overline{sid^*})$.
3) If sid^* has no matching session $\overline{sid^*}$, A asks asks either of the following questions:
   - $StaticReveal(\delta_B)$ and $EphemeralReveal(sid^*)$.
   - $StaticReveal(\delta_A)$.

- Test(sid^*): This query is in the following of the adversary's queries which in the answer of the Test query, the simulator selects a random bit $b \in_U \{0, 1\}$. If $b = 0$ the adversary obtains the session key and otherwise a random key is given to A.

Except the Test query, the adversary can continue the questions until the guess b' on b is enunciated. The adversary wins the game if the test session is fresh and $b = b'$. The advantage of the adversary is expressed by the following equation:

$$Adv\left(A\right) = \Pr\left[A\text{wins}\right] - \tfrac{1}{2}$$

ABKA Security Definition: An ABKA is secure if:

1) Two users U_A and U_B with respectively attribute sets δ_A and δ_B satisfying the predefined policy establish the same session key at the end of two matching sessions.

2) For any polynomial probabilistic adversary A, $Adv(A)$ is negligible.

The model is called *selective attribute* if the adversary declares two attribute sets δ_A and δ_B as attributes of the test session at the beginning of the security experiment.

Lagrange Interpolation:

Let a polynomial $q(x)$ of degree $d-1$ over Z_p and a set $S \subset Z_p$, $|S| = d$. Supposes the value of $q(i)$ is given for all $i \in S$. The Lagrange interpolation states that the polynomial $q(x)$ is computed as follows:

$$q\left(x\right) = \sum\nolimits_{i \in S} q\left(i\right) \Delta_{i,S}\left(x\right).$$

$\Delta_{i,S}\left(x\right)$ is the Lagrange coefficient computed as follows:

$$\Delta_{i,S}\left(x\right) = \prod_{j \in S, j \neq i} \tfrac{x-j}{i-j} \; for\ all\ i \in S.$$

3 Review of Fujioka Protocol

Let k be a security parameter and $H : \{0,1\}^* \to \{0,1\}^k$ and $H' : \{0,1\}^* \to Z_p$ be two random oracles. Let U be a set of possible attributes,γ is a access tree and $L\left(\gamma\right)$ is the set of all leaf nodes of γ. c_u is the number of child nodes of a non leaf node u which for each leaf node u, $c_u = 1$. A threshold k_u, $1 \leq k_u \leq c_u$ is assigned to each node u and for the leaf node u, $k_u = c_u = 1$. For each node u we assign $index\left(u\right) \in \{1, \cdots, c_w\}$ where w represents the parent of u. Each node u has an attribute $att\left(u\right) \in U$ that $U = \{1, \cdots, n\}$. Hence, an access tree is defined with three parameter $\left(k_u, index\left(u\right), att\left(u\right)\right)$. An attribute set δ satisfies an access tree γ as the following. A leaf node u is satisfied if $att\left(u\right) \in \delta$ and a non leaf node u is satisfied if the number of child nodes of u are at least k_u. An attribute set δ satisfies an access tree γ if the root node n_r is satisfied. The Fujioka protocol composed of three algorithms Key Generation, Key Extraction and Key Exchange as follows:

- Key Generation: This algorithm selects a random number $z \in Z_p$ and for each attribute of U, a random number $\{t_i\}_{i \in U}$ is also selected. Next it computes $Z = e(g,g)^z$ and $\{T_i = g^{t_i}\}_{i \in U}$ as public keys.

- Key Extraction: For a given access tree γ_A for the user U_A, the private key $\{D_u\}_{u \in L(\gamma_A)}$ is computed as follows: The algorithm selects a random polynomial $q_u(.)$ of degree $k_u - 1$. It sets $q_{u_r} = z$ and randomly selects the other d_{u_r} points. The algorithm sets $q_u\left(0\right) = q_{u'}\left(index\left(u\right)\right)$ for each non leaf node u in which u' is the parent of node u and the other d_u points are selected randomly and for all nodes the corresponding polynomial are created. Finally, for each leaf node $u \in L(\gamma_A)$ the private key $D_u = g^{\frac{q_u(0)}{t_i}}$ is computed where $i = att(u)$.

- Key Exchange: Let U_A and U_B be the initiator and responder of the key agreement protocol, respectively. $\{D_u\}_{u \in L(\gamma_A)}$ is the private key of U_A and $\{D_u\}_{u \in L(\gamma_B)}$ is the private key of U_B. The details of the protocol are as follows:
  - U_A with attribute set δ_A, selects a random number $\tilde{x}$ and computes $x = H'(\{D_u\}_{u \in L(\gamma_A)}, \tilde{x})$, $X = g^x$ and $\{T_i^x\}_{i \in \delta_A}$. Then he sends X, $\{T_i^x\}_{i \in \delta_A}$ and δ_A to U_B.
  - U_B with attribute set δ_B, selects a random number $\tilde{y}$ and computes $y = H'(\{D_u\}_{u \in L(\gamma_B)}, \tilde{y})$, $Y = g^y$ and $\{T_i^y\}_{i \in \delta_B}$. Then he sends Y, $\{T_i^y\}_{i \in \delta_B}$ and δ_B to U_A. Then U_B computes the session keys as follows: For each leaf node u of γ_B if $att(u) \in \delta_A$, then the following value is computed, $j = att(u)$:

$$\begin{aligned} e(D_u, T_j^x) &= e(g^{\frac{q_u(0)}{t_j}}, g^{x t_j}) \\ &= e\left(g,g\right)^{x q_u(0)} \end{aligned}$$

for each non leaf node u of γ_B, U_B defines: $\tilde{S}'_u = \{\ u_c|u_c$ is a children of u and $e(g,g)^{xq_{u_c}(0)}$isgiven$\}$. If $|\tilde{S}'_u| \geq k_u$, U_B selects a subset $\tilde{S}_u \subset \tilde{S}'_u$ that $|\tilde{S}_u| = k_u$ and sets $S_u = index(u_c|u_c \in \tilde{S}_u)$. Then the shared secrets is computed as follows, $i = index\,(u_c)$:

$$\begin{aligned}
\sigma_1 &= \prod_{u_c \in \tilde{S}_u} \left(e\,(g,g)^{xq_{u_c}(0)}\right)^{\Delta_{i,S_u}(0)} \\
&= \prod_{u_c \in \tilde{S}_u} \left(e\,(g,g)^{xq_u(i)}\right)^{\Delta_{i,S_u}(0)} \\
&= e\,(g,g)^{xq_u(0)} \\
\sigma_2 &= Z^y, \\
\sigma_3 &= X^y.
\end{aligned}$$

– U_A is the same as U_B after receiving Y, $\{T_i^y\}_{i \in \delta_B}$ and δ_B computes the shared secrets as follows:

$$\begin{aligned}
\sigma_2 &= e\,(g,g)^{yq_{u_r}(0)} \\
&= e\,(g,g)^{yz}, \\
\sigma_1 &= Z^x, \\
\sigma_3 &= Y^x.
\end{aligned}$$

Finally, the session key is computed by U_A and U_B:

$$K = H(\sigma_1, \sigma_2, \sigma_3, (\delta_A, X, \{T_i^x\}_{i \in \delta_A}), (\delta_B, Y, \{T_i^y\}_{i \in \delta_B})).$$

3.1 KCI Attack on Fujioka Protocol

Let the adversary A knows the private key of the user U_A, $\left\{D_u = g^{\frac{q_u(0)}{t_i}}\right\}_{u \in L(\gamma_A)}$ where $i = att\,(u)$. The adversary selects an attribute set $\delta_B \subset U$ such that δ_B satisfies γ_A. Afterwards A selects a random number $y \in Z_p^*$ and computes $Y = g^y$ and $\{T_i^y\}_{i \in \delta_B}$ and sends them with δ_B to U_A. U_A computes the shared secrets after receiving the massage of A according to the protocol and sends $\{T_i^x\}_{i \in \delta_A}$, $X = g^x$ and δ_A to the adversary. A compute the shared secrets after receiving the message of U_A as follows:

$$\begin{aligned}
\sigma_1 &= Z^y = e\,(g,g)^{zy}, \\
\sigma_3 &= X^y.
\end{aligned}$$

Since δ_B satisfies γ_A, there are common attributes j in δ_A and δ_B which are determined by the access tree γ_A. Therefore the adversary for all the common attributes has $\left\{e\left(D_u, T_j^x\right) = e\,(g,g)^{xq_u(0)}\right\}_{u \in L(\gamma_A)}$, where $j = att(u)$ and according to the protocol the shared secret σ_2 can be computed as:

$$\begin{aligned}
\sigma_2 &= e\,(g,g)^{xq_{u_r}(0)} \\
&= e\,(g,g)^{xz}.
\end{aligned}$$

4 The Proposed Attribute Based Key Agreement Protocol

In a bilinear pairing $e : G_1 \times G_1 \to G_2$ let $G_1$and G_2 be two cyclic groups of a prime order p and g be the generator of G_1 and $g_2 \in G_1$. Let $H : \{0,1\}^* \to Z_p$ and $H' : \{0,1\}^* \to \{0,1\}^k$ be two random oracles and $U = \{1, 2, \cdots, n\}$ be the set of possible attributes. The proposed protocol contains three algorithms; Key Generation, Key Extraction and Key Exchange as follows:

- Key Generation: This algorithm generates the master public key and the private key for PKG. The algorithm generates a random number $c \in Z_p$ and computes $g_1 = g^c$ and $e(g_2, g_1) = C$. It selects a random number $r_i \in Z_p$ for each attribute $i \in U$ and computes $g_1^{r_i}$. The outputs of the algorithm are the random numbers r_i and the master private key c. In addition, $C, g, g_1, g_2, \{g_1^{r_i}\}_{i \in U}$ are published as public parameters that C is the master public key of the PKG.

- Key Extraction: Let δ be an attribute set. The PKG uses this algorithm to create the private keys according to the attribute sets. First, the PKG selects a random polynomial $q(x)$ of degree $d-1$ that d is a threshold and $q(0)=c$. Then for each attribute $i \in \delta$, the private keys are computed as follows:

$$\begin{aligned} d_{i1} &= g_2^{q(i)} \left(H\left(i\right)\right)^{r_i}, \\ d_{i2} &= g^{r_i}. \end{aligned}$$

  The computation of the private keys imply the threshold policy of the PKG.

- Key Exchange: Let the user U_A with attribute set δ_A and the user U_B with attribute set δ_A be the two participants of the protocol. The details of the protocol are as follows:

  - U_A selects a random number $a' \in Z_p$ as the ephemeral secret key and computes $a = H\left(\{d_{i1}\}_{i\in\delta_A}, \{d_{i2}\}_{i\in\delta_A}, a'\right)$, $X = g^a$ and $X' = \left\{g_1^{-1} H\left(i\right)\right\}_{i\in\delta_B}^{a}$. Next U_A sends $\{X, X', \delta_A\}$ to U_B and deletes a'.
  - Upon receiving the message from U_A, U_B selects a random number $b' \in Z_p$ as the ephemeral secret key and computes $b = H\left(\{d_{i1}\}_{i\in\delta_B}, \{d_{i2}\}_{i\in\delta_B}, b'\right)$, $Y = g^y$ and $Y' = \left\{g_1^{-1} H\left(i\right)\right\}_{i\in\delta_A}^{b}$. U_B sends $\{Y, Y', \delta_B\}$ to U_A and deletes b'. U_B then computes the shared secrets s_1, s_2, s_3 as follows:

$$\begin{aligned} s_1 &= \prod_{i\in S_B} \left(\frac{e\left(d_{i1}, X\right)}{e\left(d_{i2}, X'\right)} \cdot e\left(g_1^{-r_i}, X\right)\right)^{\Delta_{i,S_B}(0)} \\ &= \prod_{i\in S_B} \left(\frac{e\left(g_2^{q(i)} H\left(i\right)^{r_i}, g^a\right)}{e\left(g^{r_i}, H\left(i\right)^a g_1^{-a}\right)} \cdot e\left(g_1^{-r_i}, g^a\right)\right)^{\Delta_{i,S_B}(0)} \\ &= \prod_{i\in S_B} \left(\frac{e\left(g_2^{q(i)}, g^a\right) e\left(H\left(i\right)^{r_i}, g^a\right)}{e\left(g^{r_i}, H\left(i\right)^a\right) e\left(g^{r_i}, g_1^{-a}\right)} \cdot e\left(g_1^{-r_i}, g^a\right)\right)^{\Delta_{i,S_B}(0)} \\ &= e\left(g_2, g^a\right)^{\sum\limits_{i\in S_B} q(i)\Delta_{i,S_B}(0)} \\ &= e\left(g_2, g\right)^{ac} \end{aligned}$$

  where S_B is a d-element subset of δ_B.

$$\begin{aligned} s_2 &= e(g_2, g_1)^b = e(g_2, g)^{bc}, \\ s_3 &= X^b = g^{ab}. \end{aligned}$$

  - U_A computes the shared secrets after receiving the message from U_B, as follows:

$$\begin{aligned} s_2 &= \prod_{i\in S_A} \left(\frac{e(d_{i1}, Y)}{e\left(d_{i2}, Y'\right)} \cdot e\left(g_1^{-r_i}, Y\right)\right)^{\Delta_{i,S_A}(0)} \\ &= \prod_{i\in S_A} \left(\frac{e\left(g_2^{q(i)} H\left(i\right)^{r_i}, g^b\right)}{e\left(g^{r_i}, H\left(i\right)^b g_1^{-b}\right)} \cdot e\left(g_1^{-r_i}, g^b\right)\right)^{\Delta_{i,S_A}(0)} \\ &= \prod_{i\in S_A} \left(\frac{e\left(g_2^{q(i)}, g^b\right) e\left(H\left(i\right)^{r_i}, g^b\right)}{e\left(g^{r_i}, H\left(i\right)^b\right) e\left(g^{r_i}, g_1^{-b}\right)} \cdot e\left(g_1^{-r_i}, g^b\right)\right)^{\Delta_{i,S_A}(0)} \\ &= e\left(g_2, g^b\right)^{\sum\limits_{i\in S_A} q(i)\Delta_{i,S_A}(0)} \\ &= e\left(g_2, g\right)^{bc} \end{aligned}$$

  where S_A is a d-element subset of δ_A. $s_1 = e\left(g_2, g_1\right)^a = e\left(g_2, g\right)^{ac}, s_3 = Y^a = g^{ab}$.

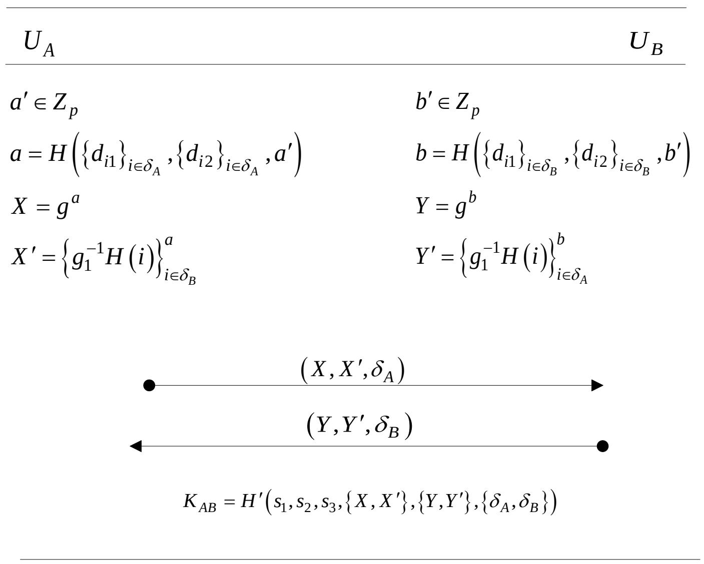

Figure 1: Details of the proposed protocol.

Finally, U_A and U_B compute the session key separately:

$$K_{AB} = H'(s_1, s_2, s_3, \{X, X'\}, \{Y, Y'\}, \{\delta_A, \delta_B\}).$$

The details of the proposed protocol is shown in the Figure 1.

5 Security Analysis

In this section, we analyze the security of the proposed protocol according to the security model described in Section 2.

Theorem: *If G_2 is a cyclic group of order of a large prime number p, and the gap BDH assumption holds and H and H' are two random oracles, then the proposed protocol is a secure attribute based key agreement protocol in the security model described in Section 1.*

Proof. We define a game between an adversary A and a simulator S utilizing the adversary to solve the gap BDH problem; whereas if the adversary can distinguish a session key of a random key then the simulator can solve the gap BDH problem. In the gap BDH problem, $U = g^u, V = g^v$ and $W = g^w$ are given to the simulator and the answer $e(U,V)^w = e(g,g)^{uvw}$ is requested. Let J be an event that the adversary asks $(s_1, s_2, s_3, \{X, X'\}, \{Y, Y'\}, \{\delta_A, \delta_B\})$ from H' and $\bar{J}$ be the complement of the event J. Let sid be the session ID of an honest user in a complete session and sid^* be a session that is not matched with sid and $sid^* \neq sid$. Let SUC be an event that the adversary succeeds and sid^* be the test session between the two users U_A and U_B, where the test session is the $j_A - th$ session of U_A. Since $sid^* \neq sid$,$\Pr\left[SUC \wedge \bar{J}\right] \leq \frac{1}{2}$ and $\Pr[SUC] = \Pr[SUC \wedge J] + \Pr\left[SUC \wedge \bar{J}\right] \leq \frac{1}{2} + \Pr[SUC \wedge J]$, where $\Pr[SUC \wedge J] \geq f(k)$, k is a security parameter and $f(\cdot)$ is a non negligible function. Hereinafter the event $SUC \wedge J$ is indicated with SUC^*. Consider the following events:

- Let D be an event that the adversary A asks $\{d_{i1}\}_{i\in\delta}, \{d_{i2}\}_{i\in\delta}$ from H. This query is before the $StaticReveal$ or $MasterReveal$ questions or it is done without the questions.
- Let $\bar{D}$ be the complement of the event D.
- Let E_1 be an event that the test session sid^* has no matching session $\overline{sid^*}$ and the adversary asks $StaticReveal(\delta_{\bar{B}})$, where $\bar{B}$ is the participant B in a session.
- Let E_2 be an event that the test session sid^* has no matching session $\overline{sid^*}$ and the adversary asks $EphemeralReveal(sid^*)$.
- Let E_3 be an event that the test session sid^* has a matching session $\overline{sid^*}$ and the adversary asks $MasterReveal$ or $StaticReveal(\delta_{\bar{B}})$ and $StaticReveal(\delta_{\bar{A}})$.

- Let E_4 be an event that the test session sid^* has a matching session $\overline{sid^*}$ and the adversary asks $EphemeralReveal(sid^*)$ and $EphemeralReveal(\overline{sid^*})$.
- Let E_5 be an event that the test session sid^* has a matching session $\overline{sid^*}$ and the adversary asks $StaticReveal(\delta_{\bar{B}})$ and $EphemeralReveal(\overline{sid^*})$.
- Let E_6 be an event that the test session sid^* has a matching session $\overline{sid^*}$ and the adversary asks $StaticReveal(\delta_{\bar{A}})$ and $EphemeralReveal(sid^*)$.
- Let E_7 be an event that the test session sid^* has no matching session $\overline{sid^*}$ and the adversary asks $StaticReveal(\delta_{\bar{A}})$.

To proof we evaluate the events $D \wedge SUC^*$ and $E_i \wedge \bar{D} \wedge SUC^*$:

1) The event $D \wedge SUC^*$:
In the event D, the adversary asks $\{d_{i1}, d_{i2}\}_{i\in\delta}$ from H and the simulator sets $e(g_2, g)^c = e(U, V)$. Since S knows the number r_i, it computes g_2^c as follows:
$g_2^c = \prod\limits_{i\in S_\delta} \left(\frac{D_i}{H(i)^{r_i}}\right)^{\Delta_{i,S_\delta}(0)} = g_2^{\sum\limits_{i\in S_\delta} q(i)\Delta_{i,S_\delta}(0)}$ Consequently the answer of the BDH problem is $BDH(U, V, W) = e(g_2^c, g^w)$.

2) The event $E_1 \wedge \bar{D} \wedge SUC^*$:
Let U_A and U_B be the two participants of the test session and δ_A and δ_B be their attribute sets, respectively. Let the test session be the $j_A - th$ session of the user U_A and $\delta_{\bar{A}}$ be the attribute set of the participant of the user U_A. In the event E_1, the session test sid^* has no matching session $\overline{sid^*}$ and the adversary asks $StaticReveal(\delta_{\bar{B}})$. According to the freshness condition of the test session A does not query $EphemeralReveal(sid^*)$ and $StaticReveal(\delta_{\bar{A}})$ or $MasterReveal$. To answer, the simulator sets $C = e(g_2, g)^c = e(U, V)$ and declares that the test session is done between the two users U_A and U_B with attribute sets δ_A and δ_B respectively. The simulator prepares three lists L_{H_1}, L_{H_2} and $L_{H'}$ to answer the queries of the random oracles H and H' and also establishes a list L_K for the query $SessionKeyReveal$. The details of the simulation are as follows:

- $H(i)$:If $i \in L_{H_1}$, S answers the recorded value in L_{H_1}, otherwise if $i \in \delta_A \backslash \delta_B$ the simulator selects a random number $\beta_i \in Z_p$ and sets $H(i) = g^{\beta_i}$ and if $i \notin \delta_A \backslash \delta_B$, S selects a random number $\beta_i \in Z_p$ and sets $H(i) = g_1 g^{\beta_i}$. Finally, S records $(i, H(i))$ in L_{H_1} and returns it.
- $H\left(\{d_{i1}, d_{i2}\}_{i\in\delta_l}, a'\right)$: If $l = A$ and a' is selected in the i_A- th session of U_A, then S aborts with failure, otherwise the simulator forms the list L_{H_2} and simulates the query as usual.
- $H'(s_1, s_2, s_3, \{X, X'\}, \{Y, Y'\}, \{\delta_l, \delta_m\})$:
  - If $(s_1, s_2, s_3, \{X, X'\}, \{Y, Y'\}, \{\delta_l, \delta_m\})$ is recorded in $L_{H'}$, then S returns the recorded value K.
  - Otherwise, if $\{X, X'\}, \{Y, Y'\}, \{\delta_l, \delta_m\}$ is recorded in L_K, $DBDH(X, U, V, s_1) = 1$, $DBDH(Y, U, V, s_2) = 1$ and $e(X, Y) = e(g, s_3)$ then S returns the recorded value K in L_K and records it in $L_{H'}$.
  - Otherwise, if $DBDH(X, U, V, s_1) = 1$, $DBDH(Y, U, V, s_2) = 1$, $e(X, Y) = e(g, s_3)$, $l = A$ and $m = B$ and the session is j_A-th session of the user U_A, the simulator stops and successfully returns $s_1 = BDH(U, V, W)$ as the answer of the BDH problem.
  - Otherwise, S returns a random value K and records in $L_{H'}$.
- $Send(I, \delta_l, \delta_m)$: If $m = B$, $l = A$ and the session is the j_A-th session of the user U_A(test session), then the simulator sets $X = W$ and $X' = W^{\beta_i}$, where β_i was recorded in L_{H_1}. Otherwise S computes X and X' as usual and based on the list L_{H_1}. Finally the simulator returns $(\delta_l, \delta_m, X, X')$.
- $Send(R, \delta_m, \delta_l, X, X')$: S computes Y and Y' according to the protocol and records $(\delta_l, \delta_m, (X, X'), (Y, Y'))$ as a complete session.
- $Send(I, R, \delta_l, \delta_m, X, X', Y, Y')$: If $(X,\ X',\ \delta_l,\ \delta_m)$ is not recorded, the simulator notifies the session $(X, X', Y, Y', \delta_l, \delta_m)$ is not complete, otherwise S records the session as a complete session.
- $SessionKeyReveal(sid)$:
  - If the session sid is not complete, S returns an error.
  - Otherwise if sid is recorded in L_K, then S returns the value K.
  - Otherwise if $(s_1, s_2, s_3, \{X, X'\}, \{Y, Y'\})$ is recorded in $L_{H'}$, $DBDH(X, U, V, s_1) = 1$, $DBDH(Y, U, V, s_2) = 1$ and $e(g, s_3) = e(X, Y)$, then S returns the recorded value K and records in L_K.
  - Otherwise S selects a random number $K \in_R \{0, 1\}^k$ and records it in L_K and returns K.

Table 1: Comparison of different protocols

	N_h	N_e	N_m	N_p	RN	FS	KSK	UKS	KCI
Fujioka Protocol[5]	1	$3+d+l$	d	d	1	+	+	+	-
Yoneyama Protocol[15]	$2l$	$3+2d+l$	$3d+2l$	$2d$	l	+	+	+	+
Our rotocol	$l+1$	$3+d+l$	$3d+l$	$2d$	1	+	+	+	+

- $StaticReveal(\delta)$: If $|\delta \cap \delta_B| \geq d$, S aborts with failure, otherwise the simulator defines a set $\Gamma = \{\delta \cap \delta_B\}$, a set Γ' with $d-1$ elements such that $\Gamma \subseteq \Gamma' \subseteq \delta$ and a set $F = \Gamma' \cup \{0\}$. Then S generates private keys d_{i1} and d_{i2} for all attributes $i \in \delta$ as follows:
  - If $i \in \Gamma'$, S sets

    $$d_{i1} = g_2^{\tau_i} \left(H\left(i\right)\right)^{r_i}$$
    $$d_{i_2} = g^{r_i}$$

    By this simulation, in fact, S defines $d-1$ points τ_i of $q(x)$ for all $i \in \Gamma'$ and also the d-th point is $q(0) = c$.
  - If $i \notin \Gamma'$, S simulates the private keys as follows:

    $$d_{i1} = g_2^{-\beta_i \Delta_{0,F}(i) + \sum\limits_{j \in \Gamma'} \Delta_{j,F}(i)\, q(j)} \left(g_1\, g^{\beta_i}\right)^{r_i'}$$
    $$d_{i2} = g_2^{-\Delta_{0,F}(i)}\, g^{r_i'}$$

    This simulation is correct because we considered $r_i = r_i' - y\Delta_{0,F}(i)$, where $g_2 = g^y$. Since $q\left(i\right) = \sum\limits_{j \in \Gamma'} q\left(j\right) \Delta_{j,F}(i) + q\left(0\right) \Delta_{0,F}(i)$, we have :

    $$g_2^{q(i)}\, H\left(i\right)^{r_i} = g_2^{-\beta_i \Delta_{0,F}(i) + \sum\limits_{j \in \Gamma'} \Delta_{j,F}(i)\; q(j)} . H\left(i\right)^{r_i'}$$
    $$g^{r_i} = g^{r_i'}\; g_2{}^{-\Delta_{0,F}(i)}$$

- $MasterReveal\ (.)$: S aborts with failure.
- $EstablishParty\ (U_i, \delta_i)$: S answers to this query as usual.
- $Test(sid)$: If $X \neq W$, then S aborts with failure, otherwise the simulator answers to this query as the definition.
- If the adversary declares the guess on b, S aborts with failure.

Let the simulation run among N users and L be the number of possible sessions. In the above simulation, the adversary selects the test session with at least probability $\frac{1}{N^2 L}$. The probability of failure in the query $Send\left(I, R, \delta_l, \delta_m, X, X', Y, Y'\right)$ is negligible, the query $MasterReveal\ (.)$ is not posed in the event E_1 and according to the query$H'\left(s_1, s_2, s_3, \left\{X, X'\right\}, \left\{Y, Y'\right\}, \left\{\delta_l, \delta_m\right\}\right)$, the adversary does not declare the guess on b. Hence, the probability that the adversary successfully solves the BDH problem is as follows:
$\Pr(S) \geq \frac{p_1}{N^2 L}$, where $p_1 = Pr\{E_1 \wedge \bar{D} \wedge SUC^*\}$.

3) The event $E_2 \wedge \bar{D} \wedge SUC^*$: In the event E_2, the test session sid^* has no matching session $\overline{sid^*}$ and according to the freshness condition of the session test, A does not ask $StaticReveal\left(\delta_{\bar{B}}\right)$ and $StaticReveal\left(\delta_{\bar{A}}\right)$ or $MasterReveal$. Since H is a random oracle, the adversary cannot obtain any information about a, unless with a negligible probability. Thus S simulates this event similar to the event $E_1 \wedge \bar{D} \wedge SUC^*$.

4) The event $E_3 \wedge \bar{D} \wedge SUC^*$: In the event E_3, the test session sid^* has the matching session $\overline{sid^*}$ and the adversary asks $MasterReveal$ or $StaticReveal\left(\delta_{\bar{A}}\right)$ and $StaticReveal\left(\delta_{\bar{B}}\right)$. According to the freshness condition of the test session A does not ask $EphemeralReveal\left(sid^*\right)$ or $EphemeralReveal\left(\overline{sid^*}\right)$. S executes key generation and key extraction steps according to the protocol, then S sets $X = g^u$, $Y = g^v$ and $s_3 = g^{uv}$ in the test session sid^*. Hence, the simulator computes $BDH(U, V, W) = e(s_3, g^w)$.

5) The event $E_4 \wedge \bar{D} \wedge SUC^*$: In the event E_4, the test session sid^* has the matching session $\overline{sid^*}$ and the adversary asks $EphemeralReveal\left(sid^*\right)$ and $EphemeralReveal\left(\overline{sid^*}\right)$. According to the freshness condition of the test session, A does not ask $MasterReveal$ or $StaticReveal\left(\delta_{\bar{A}}\right)$ and $StaticReveal\left(\delta_{\bar{B}}\right)$. Since H is a random oracle, the adversary cannot obtain any information about a, unless with a negligible probability. Thus S simulates this event similar to the event $E_3 \wedge \bar{D} \wedge SUC^*$.

6) The event $E_5 \wedge \bar{D} \wedge SUC^*$: In the event E_5 the test session sid^* has the matching session $\overline{sid^*}$ and the adversary asks $StaticReveal\,(\delta_{\bar{B}})$ and $EphemeralReveal\,(\overline{sid^*})$. According to the freshness condition of the test session, A does not ask $EphemeralReveal\,(sid^*)$ and $StaticReveal\,(\delta_{\bar{A}})$ or $MasterReveal$. Since H is a random oracle, the adversary cannot obtain any information about b and so the simulation is the same as the event $E_3 \wedge \bar{D} \wedge SUC^*$.

7) The event $E_6 \wedge \bar{D} \wedge SUC^*$: In the event E_6 the test session sid^* has the matching session $\overline{sid^*}$ and the adversary asks $StaticReveal\,(\delta_{\bar{A}})$ and $EphemeralReveal\,(sid^*)$. According to the freshness condition of the test session, A does not ask $EphemeralReveal\,(\overline{sid^*})$ and $StaticReveal\,(\delta_{\bar{B}})$ or $MasterReveal$. Since H is a random oracle, the adversary cannot obtain any information about a and so the simulation is the same as the event $E_3 \wedge \bar{D} \wedge SUC^*$.

8) The event $E_7 \wedge \bar{D} \wedge SUC^*$: In the event E_7, the test session sid^* has no matching session $\overline{sid^*}$ and the adversary asks $StaticReveal\,(\delta_{\bar{A}})$. According to the freshness condition of the test session, A does not ask $StaticReveal\,(\delta_{\bar{B}})$ or $MasterReveal$. S simulates this event similar to the event $E_1 \wedge \bar{D} \wedge SUC^*$, but at the query $StaticReveal(\delta)$ if $|\delta \cap \delta_A| \geq d$ the simulator aborts with failure. Otherwise S defines a set $\Gamma = \{\delta \cap \delta_A\}$, a set Γ' with $d-1$ elements such that $\Gamma \subseteq \Gamma' \subseteq \delta$ and a set $F = \Gamma' \cup \{0\}$. The simulator simulates the private keys d_{i1} and d_{i2} for all $i \in \delta$ the same as the event $E_1 \wedge \bar{D} \wedge SUC^*$.

□

Remark. The events $E_1 \wedge \bar{D} \wedge SUC^*$ and $E_7 \wedge \bar{D} \wedge SUC^*$ capture key compromise impersonation property because the adversary is allowed to obtain the long term private key of a party [11]. Also informally we can see that the proposed protocol is secure against KCI attack, because X' is computed by the attributes of U_B and so the adversary who obtains the long term private key of U_A can not impersonate U_B, because computing the session key without knowing the long term private key of the user U_B is impossible.

6 Conclusion

In this paper we posed a KCI attack on the Fujioka ABKA protocol and introduced a novel attribute based key agreement protocol. We formally discussed about KCI security in the proposed ABKA protocol and it is showed that our protocol is secure against the KCI attack in the random oracle model.We extended the eCK model of Fujioka et al. [5] and proved the security of the proposed protocol under the gap Bilinear Diffie-Hellman assumption.

Table 1 compares our protocol with two recent attribute based key agreement protocols. Details of the table are as follows: N_h is the number of hash functions, N_e is the number of exponents, N_m is the number of multiplications, N_p is the number of pairings, *RN* is the number of random numbers, *FS* means forward security, *KSK* means known session key property and *UKS* is unknown key security. l and d are the size of the attribute set and the threshold. The comparison shows that the proposed protocol and the Yoneyama protocol satisfy all security requirements while our protocol is more efficient than the Yoneyama protocol.

Acknowledgments

The authors would like to thank the anonymous re-viewers for their valuable comments.

References

[1] N. Attrapadung, B. Libert, and E. De Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Public Key Cryptography–PKC 2011*, pp. 90–108, Springer, 2011.
[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE Symposium on Security and Privacy (SP'07)*, pp. 321–334, 2007.
[3] J. Birkett and D. Stebila, "Predicate-based key exchange," in *Information Security and Privacy*, pp. 282–299, Springer, 2010.
[4] S. Blake-Wilson and A. Menezes, "Authenticated diffe-hellman key agreement protocols," in *Selected Areas in Cryptography*, pp. 339–361, Springer, 1999.
[5] A. Fujioka, K. Suzuki, and K. Yoneyama, "Predicate-based authenticated key exchange resilient to ephemeral key leakage," in *Information Security Applications*, pp. 15–30, Springer, 2011.
[6] M. C. Gorantla, C. Boyd, and J. M. G. Nieto, "Attribute-based authenticated key exchange," in *Information Security and Privacy*, pp. 300–317, Springer, 2010.

[7] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Automata, Languages and Programming*, pp. 579–591, Springer, 2008.
[8] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 89–98, 2006.
[9] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Provable Security*, pp. 1–16, Springer, 2007.
[10] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology–EUROCRYPT 2005*, pp. 457–473, Springer, 2005.
[11] N. P. Smart, "Identity-based authenticated key agreement protocol based on weil pairing," *Electronics letters*, vol. 38, no. 13, pp. 630–632, 2002.
[12] H. Wang, Q. L. Xu, and X. Fu, "Revocable attribute-based key agreement protocol without random oracles," *Journal of Networks*, vol. 4, no. 8, 2009.
[13] H. Wang, Q. Xu, and T. Ban, "A provably secure two-party attribute-based key agreement protocol," in *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'09)*, pp. 1042–1045, 2009.
[14] H. Wang, Q. Xu, and X. Fu, "Two-party attribute-based key agreement protocol in the standard model," in *Proceedings of the 2009 International Symposium on Information Processing (ISIP 2009)*, pp. 325–328, 2009.
[15] K. Yoneyama, "Strongly secure two-pass attribute-based authenticated key exchange," in *Pairing-Based Cryptography-Pairing 2010*, pp. 147–166, Springer, 2010.

Majid Bayat is a Ph.D. candidate in the Department of Mathematics and Computer Sciences at Kharzmi University in Tehran, Iran. He is presently a Research Assistant of Kharzmi university and Information Systems and Security Lab (ISSL) of Sharif University in Tehran, Iran. His research interests include digital signature schemes and cryptographic protocols.

Mohammad Reza Aref received the B.Sc. degree in 1975 from the University of Tehran, Iran, and the M.Sc. and Ph.D. degrees in 1976 and 1980, respectively, from Stanford University, Stanford, CA, USA, all in electrical engineering. He returned to Iran in 1980 and was actively engaged in academic affairs. He was a Faculty member of Isfahan University of Technology from 1982 to 1995. He has been a Professor of electrical engineering at Sharif University of Technology, Tehran, since 1995, and has published more than 230 technical papers in communication and information theory and cryptography in international journals and conferences proceedings. His current research interests include areas of communication theory, information theory, and cryptography.

MMOPRG Bot Detection Based on Traffic Analysis

Yuanchao Lu[1], Ye Zhu[1], Matthew Itomlenskis[1], Seetharamayya Vyaghri[1] and Huirong Fu[2]
(Corresponding author: Ye Zhu)

Department of Electrical and Computer Engineering& Cleveland State University[1]
Department of Electrical and Computer Engineering, Cleveland State University, Cleveland OH 44120-2214, USA
Department of Computer Science and Engineering& Oakland University[2]
Department of Computer Science and Engineering, Oakland University, Rochester, MI, USA
(Email: y.zhu61@csuohio.edu)
(Received Apr. 28, 2014; revised and accepted Aug. 7, 2014)

Abstract

In this paper, we investigate approaches to detect cheating with game bots in Massively Multiplayer Online Role-playing Games (MMORPGs). Cheating with game bots which can auto-play online games without human involvement is a big threat to the industry of MMORPGs. The proposed approach can detect game bots through analysis of encrypted game traffic. In the proposed detection, hidden Markov models (HMMs), known for their power in temporal pattern recognition, are trained to model game bots' gaming behaviors. A detection decision is made by testing a game trace of interest against trained HMMs. We evaluate the proposed detection approach with game traces collected from the Internet. Our experiment results show that the proposed detection can detect game bots accurately with only a small number of training traces.

Keywords: Game bot; HMM; MMORPG

1 Introduction

Given the prosperity of the Internet, the industry of online games, especially Massively Multiplayer Online Role-playing Games (MMORPGs), is growing and evolving at an incredible speed. Popular MMORPGs such as World of Warcraft [36], Age of Conan [2], Warhammer Online [34], and Ragnarok Online [27] attracted millions of gamers. With increasing popularity of MMORPGs, game bots specifically designed for MMORPGs are becoming popular.

Game bots are software applications designed to automate game play without human involvement. Game bots for MMORPGs can control and manipulate game characters in place of human players to complete time-consuming or boring tasks such as accumulating reputation or points for better tools or weapons in these games.

Cheating through game bots is a big threat to the game industry. MMORPG bots allow human players to pursue other activities while bots are playing games to accumulate resources. The unfairness problem becomes more serious for MMORPGs that allow trading resources, tools, or weapons gained in the virtual game world with real world currency. Bona fide gamers may lose interest in MMORPGs because of the unfairness and eventually quit playing MMORPGs.

The first step to stop cheating by game bots is detecting game bots. Game bot detection is not easy since in general game bots obey game rules perfectly. This task becomes more challenging when game packets are encrypted to protect gamers' privacy, since packet content is not accessible.

In this paper, we propose an approach to detect game bots based on packet timing. The detection is based on the observation that game bots' gaming behavior is different from human players' gaming behavior, e.g., human players respond to games events with different strategies according to game situations [18] while bots respond to game events with the same strategy in general. In the proposed detection approach, hidden Markov models (HMMs) are trained to model game bots' gaming behavior. A detection decision is made by testing a game trace of interest with trained HMMs.

In comparison with existing detection approaches, the proposed approach is more general since the detection is based on packet timing information only. Although we focus on the Ragnarok Online game in this paper, we believe the approach can be used to detect game bots for other MMORPGs. Another advantage of the proposed detection is zero burden on game servers and scalability: Because the proposed detection is based on analysis of the encrypted game traffic, no resources on game servers are needed. The proposed detection is scalable since it can be deployed in a distributed way, e.g., the detection can be deployed on routers.

The major contributions made in this paper can be summarized as follows:

- We propose an approach to detect game bots for MMORPGs based on encrypted game traffic. The proposed detection learns gaming behavior from training traces with hidden Markov models, a powerful tool to model temporal data.
- We evaluate the proposed detection approach with game traces collected from the Internet. Our experiment results show that the proposed detection can detect game bots accurately with only a small number of training traces.

The structure of the rest of paper is as follows: We introduce the Ragnarok Online game in Section 3. Section 2 reviews related work on game bot detection. In Section 4, we describe goals and requirements on game bot detection. Details of the proposed detection are given in Section 5. We evaluate the proposed detection with empirical experiments in Section 6. We conclude the paper and outline our future work in Section 7.

2 Related Work

In this section, we review the existing game bots and researches on detecting game bots.

Software bots are tools to automate tasks on computers or over the Internet. The examples are chatter bots [3] to automate conversation with human, click bots [9] to automate clicks on Internet advertisements, spam bots [4] to automate spamming content over the Internet, Botnets [8] to automate security attacks to networked computers, and gaming bots [1, 13, 17] to automate game play. In this paper, we focus on game bots.

Game bots are designed to farm for game resources that would otherwise take significant time or effort to obtain. Game bots have been popular for cheating in various games. The examples are Realbot [29] for Counter-Strike game, PokerBot pro [30] for online poker games, FishingBot [14] for World of Warcraft, and Openkore [24] for Ragnarok Online.

A number of efforts [5, 6, 7, 21, 32] have been carried out to detect cheating by game bots. Game bot detection based on characters' movement has been proposed in [6, 7, 21]. It has been reported that game bots move characters in a pattern quite different from human players. Support vector machine (SVM) classifier [7], subsequence analysis [21], and Bayesian classifier [6] are applied on movement data to make detection decisions. Thawonmas et al. [32] proposed bot detection approaches based on actions taken by a character of interest. In [32], detection decisions are made through support vector machine (SVM) classification based on features extracted from action data. Bot detection based on characteristics of game traffic such as response time, traffic burst, and round trip time is proposed in [5].

In this paper, we focus on detecting game bots through encrypted game traffic. Packet encryption renders most of the previous approaches ineffective or infeasible: Packet encryption prevent access to packet content so that movement and action taken by characters cannot be obtained for detections. Since the encrypted packets can be directed through anonymity networks, it is impossible to link packets so that traffic characteristics such as round trip time cannot be calculated from encrypted game traffic.

3 Ragnarok Online

In this paper, we focus on detecting game bots designed for the MMORPG, Ragnarok Online [27]. We choose Ragnarok Online mainly because of its popularity. Ragnarok Online has over seventeen million worldwide subscribers and has servers (both private and public) running all over the world. The peak concurrent users for Ragnarok Online are 800,000 and the average concurrent users are 450,000 [35].

Ragnarok Online game has a series of maps with native monsters. A screen shot of the game is shown in Figure 1. The game has a total of 39 different jobs divided into 6 categories which are Novice, First class, Second class, Transcendent first class, Transcendent second class, and Third Class [23]. There will be a chance of rebirth in the game once a player reaches the base level 99 and job level 50 [27]. A couple of jobs such as re-combat in the game are time consuming and sometimes boring. So gamers are very likely to use game bots to finish these jobs.

Figure 1: A screenshot of the ragnarok online

Table 1: Statistics of game packets from a one-hour trace of 7132 packets

Packet Type	Payload Size	Beginning 2 Bytes	Percentage
0	0	-	51%
1	2	1801	6%
2	6	1d02/b900	2%
3	7	9000/b800	1%
4	8	a700/8900	29%
5	11	8c00	4%
6	19	9001	6.4%
-	Other	-	0.6%

4 Problem Definition

4.1 Goal

Our goal is to detect cheating in online games with game bots. The typical detection scenario focused in this paper is as follows: A game system administrator with intention to defeat cheating with game bots can collect game traces generated by game bots and human players in advance. In this paper, we call these traces collected in advance as labeled traces. The administrator makes a detection decision whether a gamer is playing a game by herself or by a game bot through analysis of game traffic with knowledge learned from these labeled traces.

4.2 Requirements on Game Bot Detection

We list requirements of detecting cheating with game bots in massive player games as follows:

- Game bot detection should not affect gaming experiences. Since timing is critically important to gamers, it is desired that game bot detection should not interfere gamers' gaming experiences. Active detection approaches such as probing [22] or chatting [5] with a gamer can be effective in detecting game bots. But these active approaches generate additional packets and may possibly delay responses from gamers because of the chatting. To satisfy this requirement, we restrict our research on passive detection methods so that game bot detection causes zero interference to gaming.

- We restrict ourselves on timing-based traffic analysis for the following two reasons:

  – To protect gamers' privacy, gaming traffic can be encrypted end-to-end [25] or by directing game packets through anonymity networks such as Tor [11]. Packets are generally padded to the same length to further protect privacy. So only packet timing information is available for traffic analysis.

  – Since timing is critically important for gaming experiences, gamers usually are not willing to perturb packet timing to protect privacy.

- Game bot detection should be scalable so that it can be deployed to detect cheating in massive-player games.

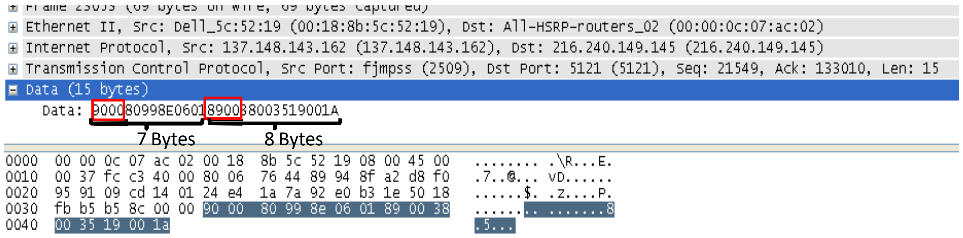

Figure 2: An example of combined packets

Table 2: Statistics of inter-packet time (statistics in rows with * are calculated without TCP ACK-only packets.)

	Rate (packet/s)	Mean of IPT (s)	Standard Deviation of IPT (s)
Walk	2.24	0.45	0.42
Fight	2.87	0.35	0.33
Talk	1.65	0.61	0.58
Walk*	1.19	0.84	1.08
Fight*	1.39	0.72	1.15
Talk*	0.91	1.11	1.23

5 Detecting Game Bots

In this section, we discuss our approach of detecting game bots in details. We first present the analysis of unencrypted game traffic generated by Ragnarok Online to explain rationales of parameter choices for HMM models[1]. Then we describe the proposed approaches to detect game bots with encrypted traffic, followed by detailed steps in game bot detection.

5.1 Analysis of Unencrypted Game Traffic

Our initial analysis of unencrypted game traffic focuses on packet size and packet timing.

Table 1 shows the statistics on the size of packets in a one-hour-long trace generated by a Ragnarok Online game client. Similar statistics can also be found in other traces. From the statistics, we can observe:

- Most packets are small. About 99% game packets have payload of length 0[2], 2, 6, 7, 8, 11, and 19 bytes. We believe small packet sizes are because of delay-sensitive nature of the online game.

- For packets of the same size, the first two bytes in payload are fixed. For example, the payload of packets with 19-bytes-long payload always starts with 9001.

- Less than 1% of game packets have payload of other length. Through checking payload of these packets, we found most of these packets are formed by combining smaller messages into one packet. One example is shown in Figure 2. The payload of the example packet is a combination of 7-byte message starting with 9000 and 8-byte message starting with 8900. We believe that these packets are generated according to Nagle's algorithm [20] which is designed to combine packets to improve network efficiency.

Similar statistics can also be found in bot traces. To find difference in bot traces and human traces, our further analysis focuses on packet timing.

The average length of inter-packet time (IPT) is 1.52s and 0.87s for client traces and bot traces respectively. The standard deviation of IPT is 5.34s and 0.75s for the client trace and the bot trace respectively. Bot traces have the smaller average and the smaller standard deviation of IPT mainly because bots responds to game events more quickly and more consistently than human players.

To demonstrate the relationship between IPT and game states, we collect a trace generated by a Ragnarok Online client when the character under control is performing different tasks in the game. Table 2 shows the distribution of IPT when the character is performing different tasks: The average IPT during attacking is much smaller than the average IPT during talking to other characters for task information and walking in maps.

The differences in the ways that human players and game bots respond to game events makes detecting game bots based on packet timing possible: (a) Human players respond to games events with different strategies according

[1]The actual game bot detection, including both training and detection phases, is based on encrypted game traffic only.
[2]The game packets without payload are TCP ACK-only packets.

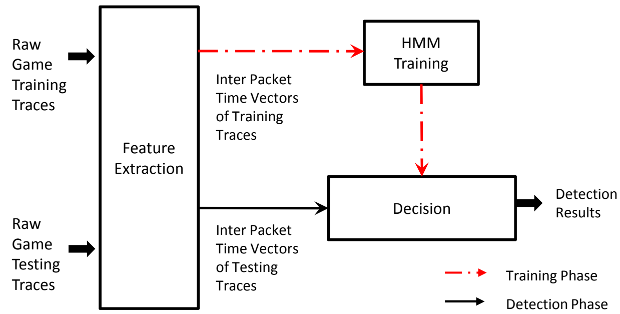

Figure 3: Steps of the proposed detection

to game situations [18] while bots respond to game events with the same strategy in general. (b) Human players can prioritize game events according to game situations and respond to game events in the order of priority. (c) In comparison with human players, bots respond to game events in a more consistent way. In the proposed detection, these differences are captured by the hidden Markov models as described in the Section 5.2. Our initial analysis also indicates: (a) Majority of packets in Ragnorak Onine game traffic can be classified into seven types according to packet size and starting bytes in payload. (b) IPT can disclose information on game states.

5.2 Detecting Game Bots with Encrypted Game Traffic

To protect gaming privacy, gamers can choose packet encryption option to encrypt game packets end-to-end [25] or by directing game packets through anonymity networks [22]. Packet encryption prevents the access to packet content by an outsider. In the mean time, packets can be padded to the same size so that no packet size information is available to outsiders.

Before introducing the proposed detection approach, we would like to briefly review the hidden Markov model (Please see [26] for an excellent introduction). The Markov Model is a tool to model a stochastic process with the Markov property that the transition from the current state to the next state depends only on the current state, i.e., independent from the past states. In a hidden Markov model (HMM), the state is not directly visible, but the output influenced by the state is observed. Each state has a probability distribution over the possible output. Therefore the sequence of the output generated by an HMM gives some information about the sequence of states. The HMM is a well-known tool to model temporal data and it has been successfully used in temporal pattern recognition such as speech recognition [28], handwriting recognition [31], and gesture recognition [10]. In the proposed detection, HMMs are trained to model gaming behaviors used for game bot detections.

For the proposed detection, we consider each pair of adjacent packets as a hidden (invisible) state. The output observation from one state is the length of inter-packet time (IPT). Since each state corresponds to an IPT, a game trace of packets $P_{s_0}, P_{s_1}, \cdots, P_{s_T}$, where P_{s_i} denotes a packet of packet type s_i[3], is a process going through T hidden states, $q_{s_0,s_1}, \cdots, q_{s_{T-1},s_T}$, where q_{s_{j-1},s_j} represents the state of the pair of adjacent packets of packet type s_{j-1} and s_j.

Ergodic HMMs [26] as shown in Figure 4 are used to model gaming behavior. We choose ergodic HMMs, in which every state of the model could be reached from every other state of the model (not necessary in one step [15]), because games are essentially loops from one game state to another game state [16]. The ergodic HMM consists of 49 states since each pair of adjacent packets is considered as a hidden state and majority of packets in Ragnorak Online game traffic can be classified into seven types as described in Section 5.1. We use $q_{i,j}$ to denote a game state of the pair of adjacent packets of packet type i and j. In the model, only transitions from a state $q_{*,j}$[4] to another state $q_{j,*}$ are allowed since two adjacent pairs of packets must share a common packet.

5.3 Detection Steps

The proposed detection method can be divided into two phases: the training phase and the detection phase as shown in Figure 3. In the training phase, the feature extraction step takes collected game traces as the input and the output vectors of IPT. HMMs are trained with these IPT vectors. In the detection phase, traces to be detected are first converted to IPT vectors through the feature extraction step. Then the converted vectors are tested against trained HMMs and detection decisions will be made based upon test results. We describe details of each step below.

[3] A list of packet types can be found in Table 1.
[4] We use * to denote a packet of any of the seven types.

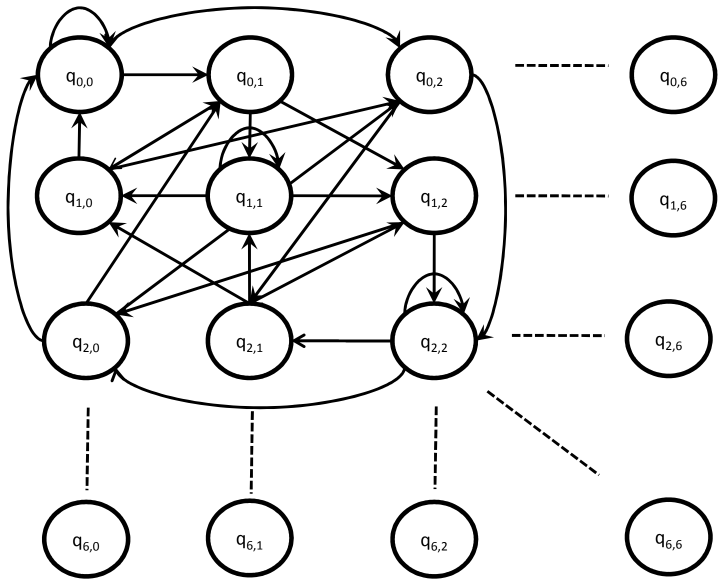

Figure 4: The hidden Markov model used in detection (to save space, only transitions between states in the upper-left corner are shown.)

5.3.1 Feature Extraction

The input of the feature extraction is the game traffic. More specifically, the input is a series of timestamps of game packets generated by human players or game bots. IPT vectors are extracted from the timestamps.

5.3.2 HMM Training

The HMM shown in Figure 4 is trained with IPT vectors generated from the feature extraction step. The HMMs to model game bots' gaming behavior can be trained by using IPT vectors extracted from game traces generated by game bots. The trained HMMs will be the output to the decision step.

The HMM training process for the detection defined in Section 4.1 can be divided into two stages: First, labeled IPT vectors generated by game bots are divided into two halves. The first half of labeled IPT vectors are used to train the HMM to model game bots' gaming behavior. In the second stage, the other half of labeled IPT vectors are evaluated against the trained HMM: A likelihood of each labeled IPT vector is calculated. Based on calculated likelihood values, a threshold T_{det} is determined. The threshold is needed in the decision step to make detection decisions. For ease of understanding, we introduce the details on threshold selection in the decision step. For the detection, the training step outputs the selected threshold and the trained HMM modeling game bots' gaming behaviors to the decision step.

5.3.3 Decision

Detection decisions for game traces of interest are made in the decision step based on knowledge learned from labeled traces. The main inputs to this step are the IPT vector generated from the game trace of interest and HMMs established in the HMM training step. The details of the decision step are described below.

For the detection, first the likelihood of the IPT vectors of interest is calculated with the HMM trained to model game bots' gaming behavior. A detection decision is made by comparing the calculated likelihood with the threshold T_{det}: If the calculated likelihood is larger than T_{det}, then the corresponding trace is declared as a trace generated by game bots. Otherwise, the trace is declared as a trace generated by human players. The rational is that a larger likelihood means that the IPT vectors of interest are more "resembling" to vectors used in training.

Obviously, the threshold T_{det} selected in the HMM training step is critical to detection performance: (a) A larger threshold can lead to a larger false negative rate, i.e., the percentage of traces generated by game bots which are detected as traces generated by human players. (b) A smaller threshold can lead to a larger false positive rate, i.e., the percentage of traces generated by human players which are detected as traces generated by game bots.

For the detection, a suitable threshold is determined in the HMM training step as follows: The second half of labeled traces generated by game bots are evaluated in terms of likelihood against the HMM trained by the first half of labeled traces. The threshold is selected so that the false negative rate (denoted as R_{fn}) on the second half

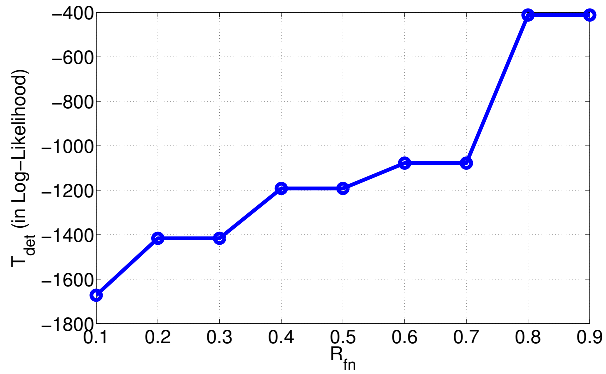

Figure 5: Threshold selected based on R_{fn}

of labeled traces is right below a predetermined threshold, say 15%. The threshold is determined solely on labeled traces generated by game bots since only labeled traces generated by game bots are available in training for the detection.

6 Empirical Evaluation

In this section, we present empirical evaluation of the proposed detection approaches. We begin this section with the description of game trace collection and performance metrics used in evaluation and then proceed with experiment results.

6.1 Data Collection

To evaluate the proposed detection approaches, we collected 38 games traces of 303,387 packets and 75.79 hours in total. Each trace is about two hours long. All game traces are collected through gaming on public Ragnarok Online game servers. Half of these traces are generated by Openkore [24], the mainstream game bot for the Ragnarok Online game. The other half of game traces are generated by human players of different proficiency levels in gaming. To evaluate the proposed detection under different traffic load, we collect game traces in rush hours, time slots with normal traffic load, and time slots with low traffic load.

6.2 Performance Metrics

We evaluate performance of the proposed bot detection with the following three performance metrics:

- **Detection Rate**: It is defined as the ratio of the number of successful detections to the number of attempts.
- **False Positive Rate**: In this paper, the false positive rate is the ratio of game traces generated by human players detected as generated by game bots.
- **False Negative Rate**: We define false negative rate as the ratio of game traces generated by game bots detected as generated by human players.

For fair comparison, IPT vectors used for training and testing are of the same size. If not specified, the IPT vectors used in training and/or testing contain 3000 IPTs. So on average, only 65.05 minutes of these two hour long game traces are used in both training and testing. If not specified, all the experiment results in the rest of this section are averaged over all possible combinations of training traces and test traces.

We present experiment results in the remaining of this section.

6.3 Detection Results

Our first set of experiments on the detection focus on the threshold T_{det}. In this set of experiments, we vary R_{fn}, the false negative rate on detecting the second half of labeled traces in the HMM training step, to select different value for the threshold T_{det}. Selected threshold values are used for detection on testing traces.

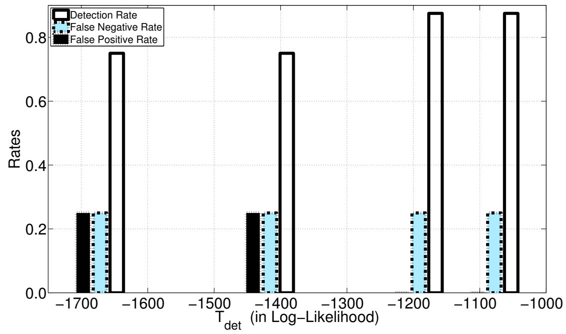

Figure 6: Detection performance with different threshold T_{det}

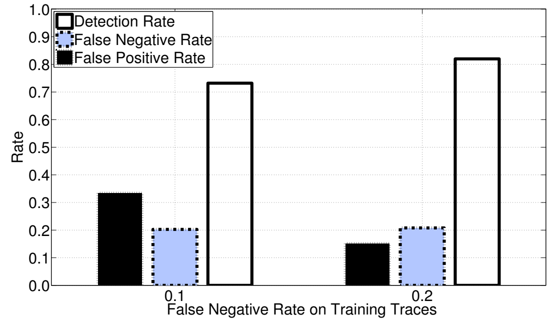

Figure 7: Detection rate vs. false negative rate on training traces in Bot detection

Typical experiment results on 10 randomly selected training traces are shown in Figure 5 and Figure 6. Figure 5 shows that the threshold T_{det} increases with R_{fn} since a larger threshold means more traces by game bots can be classified incorrectly. Detection performance on testing traces with different values of the threshold T_{det} is shown in Figure 6. We can observe that the detection rates on testing traces can approach 85% with the false negative rate and the false positive rate less than 25%. When the threshold is too large, detection rate goes down because of a larger false negative error on testing traces as expected.

Figure 7 shows performance of the detection with different R_{fn}. In this set of experiments, ten labeled traces generated by game bots are selected for training and the rest of game traces are used as testing traces. Experiment results are averaged over all possible combinations of training traces and testing traces. Figure 7 shows that with a small R_{fn}, the detection can achieve detection rates around 85% with small false positive rates and false negative rates. Our experiments also indicate that detection rate decreases when the false negative rate is too large as expected.

Our further experiments focus on length of testing vectors. In this set of experiments, we fix length of training vectors to 3,000 IPTs and vary length of testing vectors. Experiment results in Figure 8 show: (a) Better detection performance can be achieved with longer testing vectors. (b) When testing vectors are longer than 2,500 IPTs, detection rates larger than 80% can be achieved.

In summary, the proposed two bot detection techniques can achieve higher than 0.8 detection rate and the false negative rate and the false positive rate less than 0.2 respectively using less than 2,000 packets.

7 Conclusion and Future Work

In this paper, we propose an approach to detect cheating by game bots with packet timing information only. The proposed detection is based on the hidden Markov model, a powerful tool to model temporal data.

We evaluate the proposed detection approach with game traces collected from the Internet. Our experiment results show that the proposed detection can detect game bots accurately with a small number of training traces.

The HMMs used in the proposed detections do not take combined packets into account because: (1) Less than 1% of game packets are combined packets. (2) If combined packets are included in the HMMs, a large number of transitions need to be added to the HMMs and in turn more training traces are needed for training. In our

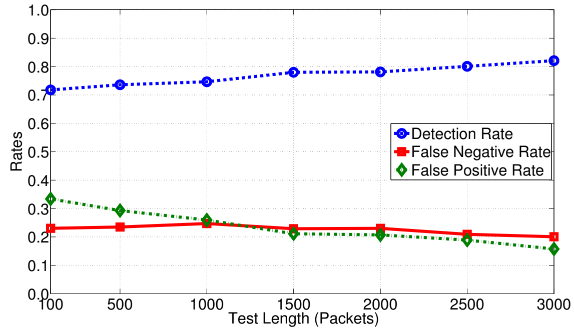

Figure 8: Detection rate vs. test length in Bot detection

experiments, we find better detection performance can be achieved without including combined packets into HMMs when less than 40 traces are available for training.

The detection approaches can be used to detect other game bots or to detect game bots for other games. Since the detection approaches are based on packet timing only. In this paper, empirical evaluation focuses on Openkore bots because other bots, such as DreamRO [12], KoreRO [19] and VisualKore [33], cannot work with recent versions of Ragnarok Online games.

To count the detection, bot designers may develop bots to delay game packets randomly or even delay game packets to emulate a human player. In our future work, we will investigate the effectiveness of the proposed approaches to detect these intelligent and possibly a new generation of game bots.

Our experiments clearly show that the proposed bot detection techniques can effectively detect game bots used in Ragnarok Online. The framework proposed in this paper includes extracting features from traffic traces and using the Hidden Markov Model to perform statistical analysis on the traces. We believe the proposed approaches can be deployed or extended to detect cheating with bots used for other purposes such as click fraud. We will investigate these extensions in our future work.

Acknowledgments

This work is partially supported by the National Science Foundation under Grant No. CNS-1338105 and CNS-1343141 and by a faculty research development grant from Cleveland State University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] R. Adobbati, A. N. Marshall, A. Scholer, and S. Tejada, "Gamebots: A 3d virtual world test-bed for multi-agent research," in *Proceedings of the Second International Workshop on Infrastructure for Agents, MAS, and Scalable MAS*, May 2001.

[2] Age of Conan. Available: http://www.ageofconan.com/

[3] A. Augello, G. Saccone, S. Gaglio, and G. Pilato, "Humorist bot: Bringing computational humour in a chat-bot system," in *CISIS'08: Proceedings of the 2008 International Conference on Complex, Intelligent and Software Intensive Systems*, pp. 703–708, Washington, DC, USA, 2008.

[4] A. Brodsky and D. Brodsky, "A distributed content independent method for spam detection," in *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, pp. 3, Berkeley, CA, USA, 2007.

[5] K. T. Chen, J. W. Jiang, P. Huang, H. H. Chu, C. L. Lei, and W. C. Chen, "Identifying mmorpg bots: a traffic analysis approach," *EURASIP J. Adv. Signal Process*, vol. 2009, no. 1, pp. 1–22, 2009.

[6] K. T. Chen, A. Liao, H. K. K. Pao, and H. H. Chu, "Game bot detection based on avatar trajectory," in *ICEC'08: Proceedings of the 7th International Conference on Entertainment Computing*, pp. 94–105, Berlin, Heidelberg, 2009.

[7] K. T. Chen, A. Liao, H. K. K. Pao, and H. C. Chang, "Game bot identification based on manifold learning," in *NetGames'08: Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games*, pp. 21–26, New York, NY, USA, 2008.

[8] D. Dagon, C. Zou, and W. Lee, "Modeling botnet propagation using time zones," in *Proceedings of the 13 th Network and Distributed System Security Symposium NDSS*, 2006.
[9] N. Daswani and M. Stoppelman, "The anatomy of clickbot.a," in *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, pp. 11–11, Berkeley, CA, USA, 2007.
[10] J. W. Deng and Ha Ta Tsui, "An HMM-based approach for gesture segmentation and recognition," in *ICPR'00: Proceedings of the International Conference on Pattern Recognition*, pp. 3683, Washington, DC, USA, 2000.
[11] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proc. of the 13th USENIX Security Symposium*, pp. 303–320, San Diego, CA, Aug. 2004.
[12] DreamRO. Available: http://www.ayxz.com/soft/1805.htm
[13] E. Drumwright and M. J. Mataric, "Generating and recognizing free-space movements in humanoid robots," in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1672–1678, Las Vegas, NV, 2003.
[14] FishingBot. Available: http://www.worldofwarcraft-gold.com/fishing-bot-world-of-warcraft-hunters/
[15] C. M. Grinstead and J. L. Snell, *Introduction to Probability.* AMS Bookstore, second edition, 1997.
[16] A. Harris, *Game Programming The L Line, The Express Line to Learning (The L Line: The Express Line To Learning).* John Wiley & Sons, Inc., New York, NY, USA, 2007.
[17] A. Khoo and R. Zubek, "Applying inexpensive ai techniques to computer games," *IEEE Intelligent Systems*, vol. 17, no. 4, pp. 48–53, 2002.
[18] H. Kim, S. Hong, and J. Kim, "Detection of auto programs for mmorpgs," in *AI 2005: Advances in Artificial Intelligence*, vol. 3809 of *Lecture Notes in Computer Science*, pp. 1281–1284, Springer Berlin Heidelberg, 2005.
[19] KoreRO. Available: http://www.kore-ro.com
[20] B. S. D. Larry, L. Peterson, Computer networks: A systems approach. 2007.
[21] S. Mitterhofer, C. Kruegel, E. Kirda, and C. Platzer, "Server-side bot detection in massively multiplayer online games," *IEEE Security and Privacy*, vol. 7, no. 3, pp. 29–36, 2009.
[22] S. J. Murdoch and G. Danezis, "Low-cost traffic analysis of tor," in *SP'05: Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pp. 183–195, Washington, DC, USA, 2005.
[23] Ragnarok online games.
"Ragnarok online games," *Gravity Interactive, Inc.*, 2002-2006.
[24] Openkore. Available: http://www.openkore.com/
[25] Phantasy Star Online: Blue Burst. Available: http://www.psoblueburst.com/
[26] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," pp. 267–296, 1990.
[27] Ragnarok Online. Available: http://www.ragnarokonline.com/
[28] C. Rathinavelu and L. Deng, "HMM-based speech recognition using state-dependent, linear transforms on mel-warped dft features," in *ICASSP'96: Proceedings of 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 9–12, Washington, DC, USA, 1996.
[29] Realbot. Available: http://realbot.bots-united.com/
[30] Bill Rini.
"Pokerbot pro - another online scam," March 2006.
[31] M. P. Schambach, "Determination of the number of writing variants with an HMM based cursive word recognition system. in *ICDAR'03: Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pp. 119, Washington, DC, USA, 2003.
[32] R. Thawonmas, Y. Kashifuji, and K. Ta Chen, "Detection of mmorpg bots based on behavior analysis," in *ACE'08: Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, pp. 91–94, New York, NY, USA, 2008.
[33] VisualKore. Available: http://mmognethood.com/node/435
[34] Warhammer Online. Available: http://eastore.ea.com/
[35] B. S. Woodcock, An analysis of MMOG subscription growth version 18.0. 2002 - 2008.
[36] World of Warcraft. Available: https://www.worldofwarcraft.com/

Yuanchao Lu received the bachelors degree from the Beijing University of Posts and Telecommunications in 2007. She is currently a graduate student in the Department of Electrical and Computer Engineering at Cleveland State University. Her research interests are on security and privacy issues of speech communications over the internet.

Ye Zhu received the BSc degree in 1994 from Shanghai JiaoTong University and the MSc degree in 2002 from Texas A&M University. He received the PhD degree from the Electrical and Computer Engineering Department at Texas A&M University. He is currently an associate professor in the Department of Electrical and Computer Engineering at Cleveland State University. His research interests include network security, and traffic engineering, wireless sensor

networks, and Vehicular Ad-Hoc Network (VANET).

Matthew Itomlenskis graduated from Cleveland State University in 2013 with a M.S. in Physics. He is currently perusing a M.S. in Computer Science at Georgia Institute of Technology while working as a Software Engineer in the field of medical imaging.

Vyaghri Seetharamayya Akella is a graduate of Cleveland State University Cleveland. He holds a Masters degree in Electrical Engineering. His research interest is in electronics and computer engineering.

Huirong Fu received her Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2000. She is currently an Associate Professor with the Department of Computer Science and Engineering, Oakland University, Rochester, MI. She joined Oakland University as an Assistant Professor in 2005. Previously, she was an Assistant Professor with North Dakota State University, Fargo, for three years, and a Postdoctoral Research Associate with Rice University, Houston, TX, for two years. As a lead professor and the principal investigator for several projects funded by the National Science Foundation, she has been actively conducting research in the areas of Vehicular Ad Hoc Networks (VANETs), information assurance and security.

Learning to Dialogue in Identifying User's Interest by Partial Observable Markov Decision Process

Oscar Li Jen Hsu, Von-Wun Soo
(Corresponding author: Oscar Li Jen Hsu)

Institute of Information Systems and Applications, National Tsing Hua University
No. 101, Section 2, Kuang-Fu Road, Hsinchu, Taiwan 30013, R.O.C.
(Email: ijeie1@olife.org)
(Received May 17, 2014; revised and accepted Oct. 5, 2014)

Abstract

A dialogue information system needs to identify the true interest of users before providing the proper information. However, how to generate a proper dialogue effectively can be difficult in identifying users true interest. In this study, an interpreter is designed to interpret user keywords input and compute the similarity between the keywords and possible users interest categories. A Partial Observable Markov Decision Process (POMDP) is used to disambiguate the uncertainty by finding a best dialogue strategy to interact with users. All the parameters in reinforcement learning in training POMDP are suggested in the method. The experimental results of three test cases in a movie information retrieval domain show that the system can apply proper dialogue strategy to interact with user and identify user interest more effectively.

Keywords: POMDP, disambiguate, mandarin, Movie Information System, word ontology

1 Introduction

A passive information retrieval usually cannot find the answer easily based merely on a simple query from the user. This is due to the hit of a query to the potential answers can result in empty, plenty or ambiguous answers in a passive information retrieval system. It is much better for the system to identify the true intent of the users before providing the answer to the query. However, to achieve such a goal, the system must have the capability to actively generate counter queries to clarify the true intents of users. In other word, it will turn the passive information retrieval system into a dialogue information retrieval system.

Many spoken dialogue information retrieval systems have been developed in such many domains such as movies [2], restaurants [3], etc. Basically, they have to deal with ambiguities due to the uncertainty of information processing from speech to text as well as text to belief.

In this study, we intend to study how the information retrieval dialogue system can generate queries to disambiguate the user's interest from words to beliefs. We focus on how a dialogue retrieval system can generate active query in order to effectively identify user's true interest. We use a simple movie retrieval problem domain as a case study. Take a query sentence in movie information retrieval for instance as in Figure 1 and Figure 2. It is assumed that the query sentence can be segmented into several words. However, the system cannot directly match the user query keyword " " with a target movie category keyword as "" in a database.

To augment the query keyword concepts in Chinese sentences, we use E-HowNet to extend the concept correlation between a keyword in the user input and the category keywords in movie database. However, when the movie dialogue information retrieval system augments the word concepts by E-HowNet interpretation, it still has a problem. The interpretation process may return several keywords that map into too many categories of movie information. To deal with this, the dialogue system may need to generate a counter query to confirm with the user.

In practice, a dialogue information retrieval system can ask the user for more information to ensure replying the user with a correct answer. But it might take many questions or options for user and cause inefficiency of retrieval. When a dialogue information retrieval system receives a variety of keywords about a movie type, there might be many possible response actions that can lead to the movie type in which the user are really interested. How can a dialogue system identify user interests based on different retrieval situations?

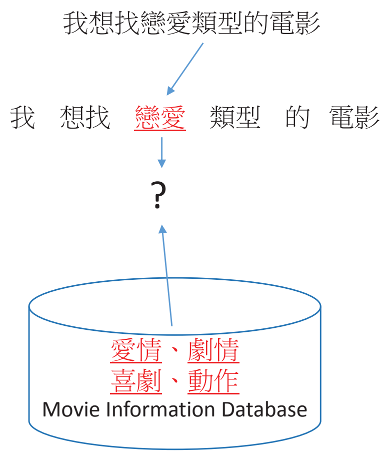

Figure 1: A user query input is segmented and the key-words are found. A keyword might not match a correct movie category.

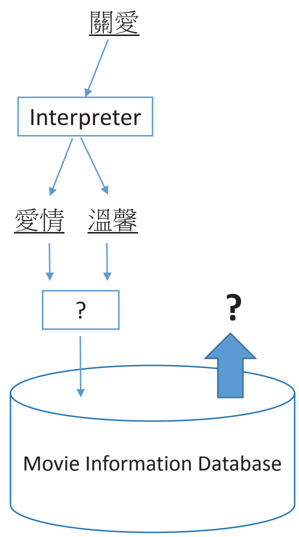

Figure 2: The interpretation process augmented by a thesaurus or ontology may or may not map into the corresponding movie categories.

- To ask for more information in order to clarify the ambiguities of user interest?
- To guess the user interest based on the likelihood after the first query from the user?

What dialogue actions to choose could effectively lead to the true interest of a user can be a challenge to the system.

We propose to use Partial Observable Markov Decision Process (POMDP) [6, 1] to solve the user interest identification problem. POMDP is expected to learn a proper policy to generate system queries that can disambiguate user interest in an effective manner. Specifically, POMDP can calculate and obtain the optimal policy that lead to the actions either:

1) To dump the movie information to user.
2) To ask the user to select possible movie type from several possible keywords.
3) To ask user to confirm a movie type of interest.

In other word, we expect POMDP to select an optimal dialogue strategy to react to users according to the initial belief states, customized rewards, and observations.

A limitation of this research is that the training results, which are strategies, in Figure 4 and Figure 5 and Figure 6 may not be 100% repeated because the random approximate method is used in this study. For example, “CONFIRM-1” in Figure 4 may become “SELECT-1,2,3” in a repeat.

The remainder of the paper is structured as follows: in Section 2, we conduct brief literature review, introduce the original and approximate POMDP method. Section 3, method, describes the design of the system, the parameters of POMDP especially, in this study. Section 4, evaluation, describes the test data and results. Section 5 is the conclusion in this study.

2 The Literature Review

In this section, a traditional POMDP method [6, 7] and an approximate POMDP method [8] are introduced. The traditional formulas shows how the optimal strategy is found by POMDP and explains the underlying ideas of a POMDP. However, the performance of the traditional formulas is too poor to be of practical use. The approximate method, PERSEUS, shows a feasible method which is implemented in this study.

2.1 The traditional POMDP formula

The reinforcement learning of POMDP is expressed in terms of formula (1), which is a value update formula of utility score on user interests in terms of a probability distribution over all possible states under a given a policy p. The capital P represents a vector of all policies p's, and $a(p)$ represents an action a given the policy p. The value function V_p as a vector over all possible belief states combines a given immediate reward $R_{a(p)}$ with a discounted reward based on observations O_a with discount rate γ while carrying the action a under policy p; T_a is a transition matrix which records the transition probabilities from current states to next states for carrying out an action "a(p)".

The observation "o" in V_{p_o} in (1) is a divergence path in a policy tree "p". In another word, it is still the same policy tree with $\overrightarrow{V_p}$.

$$\overrightarrow{V_p} = R_{a(p)} + \gamma \cdot T_{a(p)} \cdot O_{a(p)} \cdot \overrightarrow{V} \ , \ \overrightarrow{V} \equiv \begin{bmatrix} V_{p_{o1}} \\ V_{p_{o2}} \\ \vdots \end{bmatrix} \tag{1}$$

In (2), ϕ^* is the best strategy for interacting with user, which is learned by POMDP module. Those parameters are defined in the method of this study.

$$\phi^* = \arg\max_{p \in P} \beta \cdot \overrightarrow{V_p} \tag{2}$$

Given the reward values and observation scores, POMDP will learn to maximize the expected reward and obtain an optimal policy graph for action selection in response to a variety of inputs and belief states. However, the traditional approach of computing POMDP policy graph can become intractable when the problem size becomes large. The approximate sampling methods of estimating the value function of POMDP were proposed.

2.2 An approximate POMDP method - PERSEUS

Spaan and Vlassis [8] showed a way to obtain the separate belief vectors from the value iteration, called "backup". They used about ten thousand samples of belief vectors to estimate the value function. The way they generated sample beliefs is to find the successor belief b_{n+1} from current belief b_n according to the Bayes' rule while the action, the next state and the observation are randomly chosen. An action is chosen according to a uniformly probability distribution; the next state is chosen according to the probability in the transition matrix of the action; and an observation is chosen according to the probability in the observation matrix of the action and the next state. The belief sampling process starts with finding a successor of the initial belief that could be a uniform belief. The process will keep finding a sequence of successor beliefs by finding a successor of the successor belief repeatedly until the total number of beliefs reach the given required number. The approximate method of PERSEUS can improve the training time and make POMDP method more feasible. Because Algorithm 1 would be performed many times, more than one hundred rounds, to terminate, we can limit the training time or the number of repeat times as in practice, to get a good-enough result that approaches to the optimal. Equations (3), (4), (2.2), (5) define "backup()".

$$backup(b) = \alpha_{n+1}^b = argmax_{\{g_a^b\}_{a \in A}} b \cdot g_a^b \tag{3}$$

$$g_a^b = r_a + \gamma \sum_o argmax_{\{g_{a,o}^i\}_i} b \cdot g_{a,o}^i \tag{4}$$

Algorithm 1 PERSEUS backup stage: $V_{n+1} = \tilde{H}_{PERSEUS} V_n$

Require: V_n
Ensure: V_{n+1}

```
Set V_{n+1} = ∅.
Initialize B̃ ← B.
repeat
    Sample a belief point b uniformly at random from B̃
    Compute α = backup(b).
    if b · α ≥ V_n(b) then
        add α to V_{n+1}
    else
        add α' = argmax_{{α_{n,i}}_i} b · α_{n,i} to V_{n+1}.
    end if
    Compute B̃ = {b ∈ B : V_{n+1(b)} < V_n(b)}.
until B̃ = ∅
```

$$\overrightarrow{g^i_{a,o}} \equiv \begin{bmatrix} g^i_{a,o}(s1) \\ g^i_{a,o}(s2) \\ \vdots \end{bmatrix}, \quad g^i_{a,o}(s) = \sum_{s'} p(o|s', a) p(s'|s, a) \alpha^i_n(s').$$

$$\alpha_0 = \{\overrightarrow{v}\}, \quad \forall x \in \overrightarrow{v}, x = \frac{min(R)}{(1-\gamma)} \tag{5}$$

In (5), γ is discount factor, R represents all of the reward values in POMDP. In (5), the gain function $g^i_{a,o}(s)$ at state s at the iteration i is computed by combining by a old value function α with all probability sensor models $p(o|s', a)$ and action models $p(s'|s, a)$ over all possible transition state s' given observation o while carrying out action α at state s. In (4), the gain function g^b_a is computed by combining reward r given action α with the discounted expected gain function value that is summed over all possible observations. In (3), the backup value is the best gain function value g^b_a that yields the maximum of the expected gain function values given belief states b over all possible action α.

The evaluation results of PERSEUS shows, as compared with other method, a better control quality including terms of high expected reward and less training time.

3 The Methods

In Figure 3, we show an overall information flow the dialogue movie information retrieval system. E-HowNet Module will output a belief vector that consists of probabilities of eighteen movie types after the semantic processing and inference of user's query word. POMDP module determines a best strategy to interact with the user by the "SELECT" or "CONFIRM" queries. After identifying the user interest, POMDP module outputs a word of a movie type to Movie Database System. Finally, a word of a specific movie type is entered to the movie database system and into the "WHERE" condition of a "SQL" command. The movie database system dump the movie information of a specific type to user.

To augment the keyword concept using word similarity in terms of the shortest semantic distance in an ontology model is a common way [6, 5]. In an ontology model, words are organized into a conceptual/semantic hierarchical tree in such a way that the semantic distance between two words can be defined as the length in the path between two word concepts in the semantic hierarchy in the model. In this study, E-HowNet is used as the ontology model.

A sequential decision problem for a fully observable, stochastic environment with a Markovian transition model and additive rewards is called a Markov Decision Process (MDP). When the environment is only partially observable with hidden states, Partially Observable MDPs (POMDPs) are used [9]. We formulate the dialogue generation as a sequence decision problem in which an information retrieval dialogue system must decide a best query policy to find a query action in resolving the ambiguities in the dialogue and provide answer as efficient as possible to the user. POMDP is a powerful tool that has been used for many applications [1]. For Distributed Database Queries, they used POMDP to speed up time between user information query and response on the Internet with distributed database. For Marketing, POMDP made a strategy to talk for finding potential customer for salesperson. In this study, POMDP is used to find a potential movie type by interacting with user, and made the interactions as few as possible to speed up the query and response process.

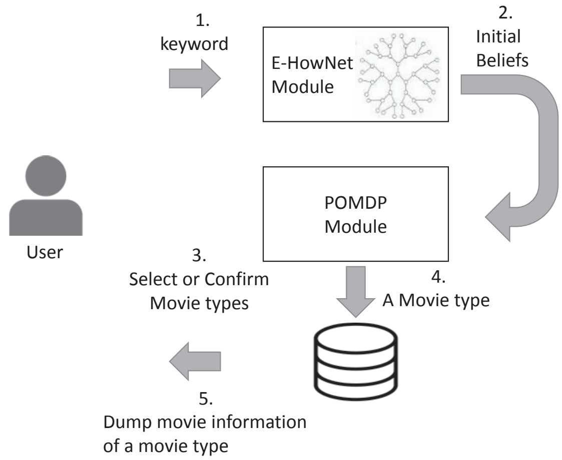

Figure 3: The information flow of the dialogue movie information retrieval system.

3.1 E-HowNet Module

This study used eighteen movie types that are defined from a movie website as POMDP states as shown in Table 1. The movie types are transformed into ontology nodes manually as Table 2. One movie type might be presented by multiple nodes. In this study, the system choose the one with the shortest path from a keyword ontology node.

The specific algorithm with E-HowNet is shown in Algorithm 2 that will generate an initial belief states vector for the use of the POMDP module.

Table 1: The list of eighteen movie types from a movie website

Movie Types in Chinese and English								
愛情 Love	音樂/歌舞 Musical	劇情 Drama	動畫 anime	紀錄片 doc	喜劇 comic	懸疑 Mystery	驚悚 Thrill	動作 action
奇幻 Fantasy	溫馨 warm heart	科幻 Sci-Fi	家庭 Family	恐怖 Horror	犯罪 Crime	冒險 adventure	戰爭 War	勵志 Inspire

Algorithm 2 E-HowNet Module

Require: keyword κ

Ensure: InitialBeliefVector $\overrightarrow{\beta}$

Begin

kNodes $\leftarrow$ to convertFromEHowNet(κ)

$\overrightarrow{\mathbf{depth}} \leftarrow$ getDepthFromCommonAncestor({ontology nodes of eighteen movie types}, kNodes).

sort($\overrightarrow{\mathbf{depth}}$)

$\overrightarrow{\mathbf{Prob}} \leftarrow 0.5e^{-0.5\cdot\overrightarrow{\mathbf{depth}}}$

$$\overrightarrow{\beta} \leftarrow \frac{\overrightarrow{\mathbf{Prob}}}{sum(\overrightarrow{\mathbf{Prob}})}$$

End

In Algorithm 2, E-HowNet module gets a keyword κ at the beginning. κ is converted into one or more ontology nodes according to E-HowNet. The algorithm calculates the depth from an ancestor to κ nodes and movie type

Table 2: movie types map to ontology nodes

Types	Ontology Nodes
愛情	'emotion| 情感', 'love| 愛戀'
音樂/歌舞	'music| 音樂', 'perform| 表演', 'sing| 唱'
劇情	'describe| 描寫', 'shows| 表演物'
動畫	'image| 圖像', 'shows| 表演物', 'draw| 畫'
紀錄片	'record| 記錄'
喜劇	'interesting| 趣'
懸疑	'uneasy| 不安'
驚悚	'fear| 害怕'
動作	'fight| 爭鬥'
奇幻	'queer| 怪'
科幻	'scientific| 科學'
溫馨	'happy| 幸福'
家庭	'child| 少兒', 'family| 家庭'
恐怖	'fear| 害怕'
犯罪	'law| 律法', 'guilty| 有罪', 'vicious| 罪惡'
冒險	'venture| 冒險'
戰爭	'military| 軍', 'fight| 爭鬥'
勵志	'cultivate| 培養', 'urge| 促使'

ontology nodes. The ancestor is the youngest common ancestor among κ and a movie type. For each movie type, the smallest depth is taken. In other words, there are eighteen depth values and they are organized into a vector. This complex depth calculation method is because an ontology node may have an unfair depth value in the multilevel classification. The elements of the depth vector and corresponding movie types are sorted from small to large depth. This operation is because the "types" in "SELECT" action in Table 3 is also sorted and it reduce the number of actions and forces the POMDP module to start an interaction from a movie type with the highest probability. The depth vector is converted into a probability vector according to Exponential distribution as (6)

$$f(\vec{v}) = 0.5e^{-0.5\vec{v}}, \qquad \forall x \in \vec{v}, x \geq 0. \tag{6}$$

Finally, the vector is normalized in which each element is divided by the summation of all eighteen elements in the vector. The final result is the initial probability belief vector β.

3.2 POMDP Module

In this section, parameters of POMDP are implemented according to the movie domain problem. The eighteen movie types are presented as eighteen states in POMDP. The information retrieval dialogue system in this study has a set of actions to response to the user according to different dialogue situations. The actions list in POMDP is shown in Table 3. Each action family represents a set of subactions according to its argument parameters.

Table 3: The list of actions

Action family	Show Message	# sub-action
SQL(Movie_Type)	Dump the movie information about (Movie_Type)	18
SELECT(types)	" 您想找的影片類型是下列哪一個呢?"	153
CONFIRM(Movie_Type)	" 您想找 (Movie_Type) 類的電影嗎?"	18

The first action name "SQL(Movie_Type)" represents that database dumps all movie information for a single specific "Movie_Type". The "SQL" is a simple SQL command in the database management system and "WHERE" condition is filled with "type = (Movie_Type)". This action is actually a termination action because the system will end the dialogue after this action.

The "SELECT" action query for the system is expressed as "SELECT(types)" which is to ask the user to choose from a list of types where types $\in \{A \sim B | A \in \{1, 2, .., 17\}, B \in \{2, 3, ..., 18\}, A < B\}$. For example, types $\{1 \sim 3\}$ = {1,2,3}. "SELECT({1,2})" means to ask the user to choose one of two options including movie type 1 and movie type 2, for instance, which may represent "", a love story, or "", an action film. Finally, "CONFIRM(Movie_Type)" is to ask the user to confirm if the "Movie_Type" is user's interest.

An observation is that the user feedback is observed as "Movie_Type" in response to the query actions that the system asked. The dimensions of all observation matrices are 18×18. The setting of observation matrix is expressed in terms of formulas (7), (8), and (9). Each column represents a "Movie_Type" from "SELECT(types)" or "CONFIRM(Movie_Type)" and there are eighteen column. The setting of reward is given as formulas (10), (11), (12) in which each row is related to a state and each column is related to a observation.

$$O_{SQL(Movie\_Type)} \equiv \mathbf{1}_{18\times 18} \times \frac{1}{18} \tag{7}$$

Equation (7) represents eighteen matrices which are just reset the dialogue and no need to detect any observations further.

$$O_{SELECT(\tau)} \equiv \begin{bmatrix} O_{SE(\tau)_{1,1}} & \cdots & O_{SE(\tau)_{1,18}} \\ \vdots & \ddots & \vdots \\ O_{SE(\tau)_{18,1}} & \cdots & O_{SE(\tau)_{18,18}} \end{bmatrix},$$
$$O_{SE(\tau)_{i,j}} \equiv \begin{cases} 1, & \text{if} \quad i \in \tau \ and \ j = i \\ 0, & \text{if} \quad (i \in \tau \ and \ j \neq i) \\ & \quad\quad or \ (i \notin \tau \ and \ j \in \tau) \\ \dfrac{1}{18 - |\tau|}, & \text{if} \quad i \notin \tau \ and \ j \notin \tau \end{cases} \tag{8}$$

Equation (8) represents one hundred and eighty-nine matrices correspond with the number of actions "SELECT". τ is "types" in Table 3. The entries of these matrices represent the match with the states that the user chooses. If there is match then the score is as high as 1, or 0 if no match. Otherwise, values will uniformly distribute over other states that is not appear in τ.

$$O_{CONFIRM(\mu)} \equiv \begin{bmatrix} O_{CO(\mu)_{1,1}} & \cdots & O_{CO(\mu)_{1,18}} \\ \vdots & \ddots & \vdots \\ O_{CO(\mu)_{18,1}} & \cdots & O_{CO(\mu)_{18,18}} \end{bmatrix}$$
$$O_{CO(\mu)_{i,j}} \equiv \begin{cases} 1, & \text{if} \quad i = \mu \ and \ j = \mu \\ 0, & \text{if} \quad i = \mu \ XOR \ j = \mu \\ \dfrac{1}{17}, & \text{if} \quad i \neq \mu \ and \ j \neq \mu \end{cases} \tag{9}$$

Equation (9) represents eighteen matrices. μ is a movie type. The entries of these matrices represent values will reflect the score of a state μ after user accepts it. Otherwise, the value of state μ will uniformly distribute over other seventeen states.

$$R_{SQL(\mu)} \equiv \begin{bmatrix} R_{SQL(\mu)_1} \\ R_{SQL(\mu)_2} \\ \vdots \\ R_{SQL(\mu)_{18}} \end{bmatrix},$$
$$R_{SQL(\mu)_i} \equiv \begin{cases} 15, & \text{if} \quad i = \mu \\ \dfrac{-10}{17}, & \text{if} \quad i \neq \mu \end{cases} \tag{10}$$

$$R_{SELECT(\tau)} \equiv \overrightarrow{1}_{18} \times -|\tau| \tag{11}$$

$$R_{CONFIRM} \equiv \overrightarrow{1}_{18} \times -1 \tag{12}$$

The number of reward vectors in Equations (10), (11), and (12) are eighteen, one hundred fifty three, and eighteen, respectively. In our case, in Equation (10), it heuristically gives fifteen points to a right state and total negative ten points to all of the other wrong states. However, the concept of right and wrong does not mean that POMDP module already knows what the user interest is before the interaction or calculation. The positive reward and the negative reward for POMDP module simply represent how much merit and cost that the user might want and don't want for the given query information. Based on the intuition, it is easier to understand the phenomena that more positive reward is given, more tendency the POMDP module will to ask the information for a specific state based on a belief. In contrast, a negative reward is given, the less tendency of the POMDP for a query action. POMDP module will ask more questions or more options in "SELECT". In Equation (11), τ is as "types" in Table 3 and $|\tau|$ is the cardinality of τ. Equation (11) represents that some negative reward will be given by asking one "SELECT" question. The more options are presented in the "SELECT", the more is negative reward given. In Equation (12), it represents that one negative point will be given by asking one "CONFIRM" question. It should be noticed that the range between the positive reward points and the negative reward points in Equation (10) should not be too big because it will make the cost between Equations (11) and (12) to be relatively small and POMDP module will then generate a poor strategy.

$$T_{SQL} \equiv \mathbf{1}_{18\times 18} \times \frac{1}{18} \tag{13}$$

(13) represents eighteen matrices.

$$T_{SELECT} = T_{CONFIRM} \equiv \mathbf{I}_{18\times 18.} \tag{14}$$

In (14), T_{SELECT} represents one hundred and thirty-nine matrices. $T_{CONFIRM}$ represents eighteen matrices .

In Equation (13), the uniform matrices represent that the dialogue will be reset after a "SQL" action is given. It denotes that "SQL" actions are terminal actions while information is given to user and there is no need to ask further question. In Equation (14), the identity matrices represent that a state would not transpose into other states. In contrast, this study focuses on beliefs. The probabilities in a belief vector should concentrate on a single state when the user's interest is found. The discount factor γ is 0.95 the same as many other previous POMDP model setting [4].

The parameters above, Initial Beliefs, Actions, States, Observations, Transition matrix, Rewards, are given to POMDP that will learn to maximize the expected reward and store an optimal policy graph for action selection in response to a variety of inputs and belief states.

4 The Evaluation and Discussion

We have done several tests and shown the expected rewards for comparing the control quality. The mainframe of test environment in this study is as below.

- memory: DDR3-1333 28GB with ECC;
- CPU: Xeon 1230 v2;
- OS: Windows 7, 64 bits;
- Matlab R2013a 64 bits.

Table 4 shows results of three tests using different ontology depths as examples. The "Depth" vector is mapped from ontology, where each element in the vector is the depth between a keyword and a movie type. The "initial belief" is a probability vector which is transformed from "Depth". "Depth" and "initial belief" are both described in Algorithm 2. The "sample beliefs" is a parameter to indicate how many beliefs are to be sampled and entered into PERSEUS iteration for training which is described in Section 2.2. "Training time" indicates how long it can take to get the final strategy. Only the "depth" in the first case is generated by keyword "R" and the elements are mapped to a movie type vector as below:

The "depth" vectors in other cases are generated manually for evaluation of whether the final strategy works at different situations.

[愛情, 家庭, 恐怖, 冒險, 懸疑, 驚悚, 劇情, 犯罪, 音樂/歌舞, 動畫, 科幻, 喜劇, 奇幻, 戰爭, 勵志, 溫馨, 動作, 紀錄片 $]^T$

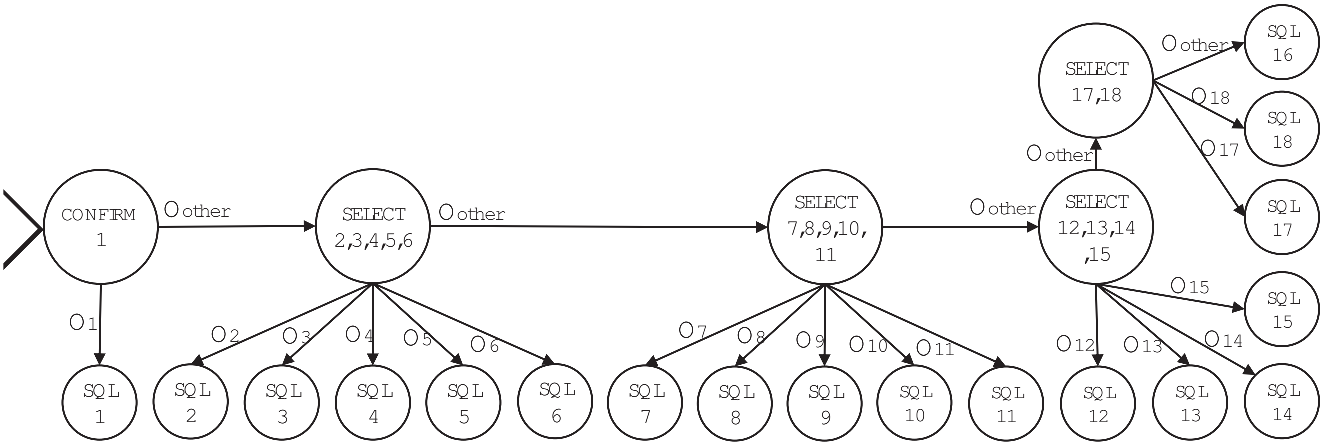

Figure 4: The policy graph obtained by 1st test.

Figures 4, 5, 6 are policy graphs generated by POMDP module under three different tests respectively. Eighteen actions from "SQL1" to "SQL18", represent that the system dumps movie information from type 1 to type 18 respectively. In Figure 4, "CONFIRM-1" action represents that the system asks the user to confirm movie type 1, which is a "yes/no" question. "SELECT_2,3,4,5,6" action represents that the system asks the user to select one movie type out of types 2, 3, 4, 5, and 6. "O2" represents an observation that the user selects a movie type 2, and so on, but "Oother"represents that the user rejects all the available options. Figure 7 shows an interface and a user interaction process of test 1. System interacts with user by the strategy of test 1 until the user finds what s\he want or until a final movie type is provided. In this example, the user denies all of the options. In reality, the user might accept the first question and end the interaction. Figure 8 shows an interaction without any POMDP strategy. It's shown that Figure 7 has a better interaction process. The user gets maximal expected utility using least number of options and questions.

By Equation (15), the expected reward can be gotten for the evaluation in this study. Each R relates to a question round indexed by R(j,p) which is a recursive function where j is the j-th question being asked. The initial value of j is "1" which represents the first round. The initial value of p is "0" and "1-p" represents the probability of j-th round. B_{other} is a belief of a type that is not in t. tj is the set of belief types in j-th question. In test 1, $B_{other} = 0.0244$ which is the 16th belief in the belief vector. γ, a discount factor, is 0.95. t_j is a set the types in the j_{th} question. $|t|$ is a total number of questions being asked. In test 1, for example, t = {1, {2,3,4,5,6}, {7,8,9,10,11}, {12,13,14,15}, {17,18}}. B_i is i_{th} belief in a belief vector. ρ, a goal reward, is 15 which also the same as discussed in the method.

$$\begin{aligned} R(j,p) = (1-p)\cdot(-|t_j|) + \rho\cdot(\sum_{i\in t_j} B_i) + \\ \gamma\cdot R(j+1, p+(\sum_{i\in t_j} B_i)), \forall j \in 1,2,...,|t|-1 \end{aligned} \quad (15)$$

$$R(j,p) = (1-p)\cdot(-|t_j|) + \rho\cdot(\sum_{i\in t_j} B_i + B_{other}), where\ j = |t|$$

$$ExpectedReward = R(1,0)$$

In Figure 8, suppose the system simply asks a question with eighteen options which deserve -18 negative reward. The total probability of all eighteen types is 1 and the user interested movie type can receive 15 reward points. Therefore, the expected reward in the case of Figure 8 is -3.

$$-18 + 1 \times 15 = -3.$$

Table 4: Three test cases

#	1		2		3	
	Depth	initial belief	Depth	initial belief	Depth	initial belief
	0	0.2968	0	0.1989	1	0.0555
	3	0.0662	0	0.1989	1	0.0555
	3	0.0662	1	0.1207	1	0.0555
	3	0.0662	1	0.1207	1	0.0555
	3	0.0662	2	0.0731	1	0.0555
	3	0.0662	2	0.0731	1	0.0555
	4	0.0402	3	0.0443	1	0.0555
	4	0.0402	3	0.0443	1	0.0555
	4	0.0402	4	0.0269	1	0.0555
	4	0.0402	4	0.0269	1	0.0555
	4	0.0402	5	0.0163	1	0.0555
	4	0.0402	5	0.0163	1	0.0555
	4	0.0402	6	0.0099	1	0.0555
	5	0.0244	6	0.0099	1	0.0555
	5	0.0244	7	0.0060	1	0.0555
	5	0.0244	7	0.0060	1	0.0555
	7	0.009	8	0.0036	1	0.0555
	7	0.009	8	0.0036	1	0.0555
sample beliefs	10000					
Training time (hour)	1					

The result showed that all of the three test case has higher expected reward than a simple interaction Evaluation result by formula (15) shows that all of the three test cases obtained a higher expected reward than the simple interaction case as in Figure 8.

5 The Conclusion

In a dialogue information retrieval system, keywords extracted from user's query may not directly match the actual words of the target information in the database. Sometimes, there can be ambiguous when user's keyword maps into multiple words. In these cases, the dialogue information need to resolve the ambiguities by actively generating some queries. However, the queries generated by the dialogue system will hopefully to be effective. In our case of applications on movie information retrieval dialogue system, we need to find a balance between to directly give a movie type with the highest probability and to ask user dozens of questions and options to identify the user's true interest. This study adopts POMDP to learn to conduct a dialogue to identify user's interest in movie types even if the user query might be ambiguously specified. A movie retrieval problem is formulated as a case study. When a movie type keyword is entered into the system in this study, the keyword is mapped to eighteen movie types by searching E-HowNet ontology model. The "E-HowNet Module" algorithm that design for this study shows a basic method of mapping words and calculating initial beliefs. The ontology nodes of eighteen movie types are described for further computation. POMDP method is used for analyzing a situation and a best interaction strategy to find

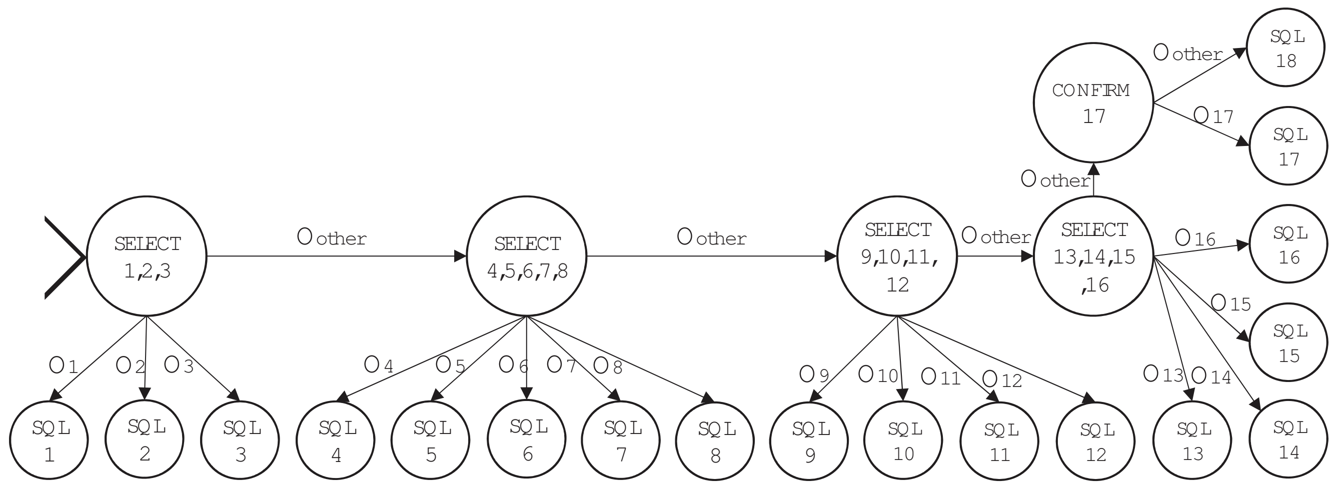

Figure 5: The policy graph of 2nd test.

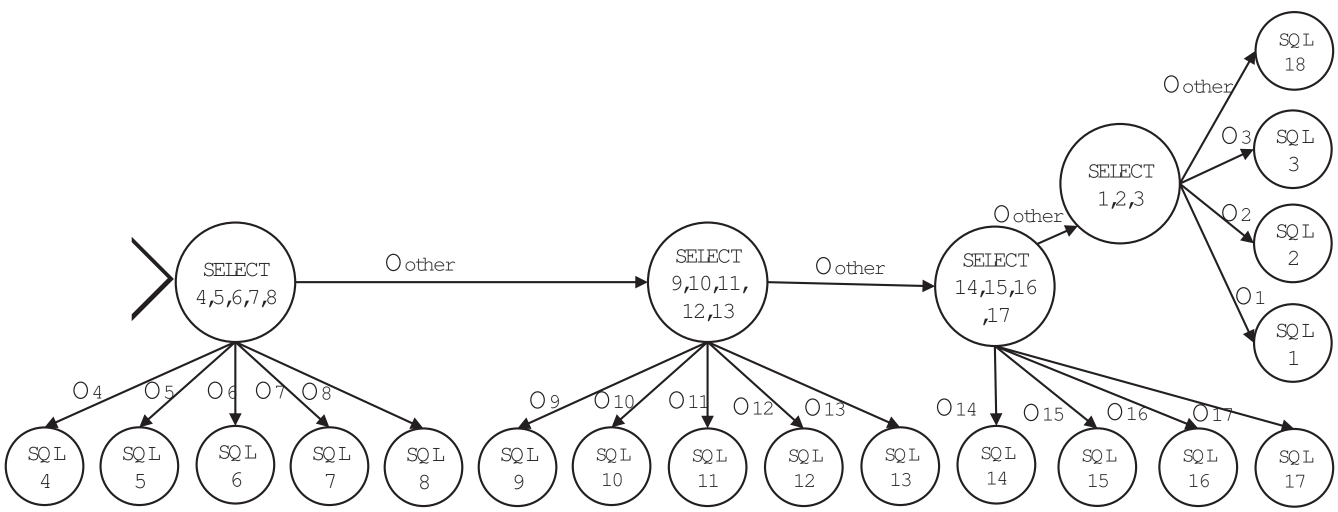

Figure 6: The policy graph of 3rd test.

out user interest will be given. POMDP seems to be feasible to suggest a proper strategy to interact with users that minimizes the number of queries and selection options. The model for POMDP includes one hundred and eighty-nine actions, eighteen observations, eighteen states. The scaled up problem can make the POMDP learning become intractable. The PERSEUS approximate method is recommended to replace the traditional POMDP formulas for a feasible training speed. PERSEUS reduced the training time to one hour. It is much faster than traditional POMDP learning method of finding an optimal policy graph that crashed before it reached a solution. The result shows that all of the three test cases under different initial conditions ended up with higher expected rewards than the case in Figure 8 which simply asks all options of movie types. In that sense, POMDP module does give a better acceptable interaction strategy.

Acknowledgments

The E-HowNet is provided by Chinese Language and Knowledge Processing Group, Academia Sinica.

系統：請輸入您想查詢的電影類別?
使用者：吃醋
系統：您想找”愛情”類型的電影嗎?

1.Yes 2.No

使用者：2
系統：您想找的影片類型是下列哪一個呢?

2. 家庭 3. 恐怖 4. 冒險 5. 懸疑 6. 驚悚 7. 都不是

使用者：7
系統：您想找的影片類型是下列哪一個呢?

7. 劇情 8. 犯罪 9. 音樂/歌舞 10. 動畫 11. 科幻 12. 都不是

使用者：12
系統：您想找的影片類型是下列哪一個呢?

12. 喜劇 13. 奇幻 14. 戰爭 15. 勵志 16. 都不是

使用者：16
系統：您想找的影片類型是下列哪一個呢?

17. 動作 18. 紀錄片 16. 都不是

使用者：16
系統：”溫馨”類的電影如下

名稱: 愛回來
上映日期: 2014-04-25
片長: 01:38
演員:《舞孃俱樂部》艾倫康明、《殺戮行動》葛瑞特迪拉杭特
劇情簡介: 一個關於愛、包容和家庭的動人故事，改編自真人真事。一名唐氏症少年遭受... 詳全文

名稱: 重點是，我愛你
上映日期：2014-02-07
片長:
演員: 歐亨尼奧德伯茲 (Valentin)、Jessica Lindsey
劇情簡介: 瓦倫廷，是阿卡普爾科玩世不恭的花花公子，直到一名妙齡女子莫名地... 詳全文

...

Figure 7: A dialogue generated by the policy graph in Figure 4

系統：請輸入您想查詢的電影類別?
使用者：吃醋
系統：您想找的影片類型是下列哪一個呢?

1. 愛情 2. 家庭 3. 恐怖 4. 冒險 5. 懸疑 6. 驚悚
7. 劇情 8. 犯罪 9. 音樂/歌舞 10. 動畫 11. 科幻
12. 喜劇 13. 奇幻 14. 戰爭 15. 勵志 16. 溫馨
17. 動作 18. 紀錄片

使用者：16
系統：”溫馨”類的電影如下

名稱: 愛回來
上映日期: 2014-04-25
片長: 01:38
演員: 《舞孃俱樂部》艾倫康明、《殺戮行動》葛瑞特迪拉杭特
劇情簡介: 一個關於愛、包容和家庭的動人故事，改編自真人真事。一名唐氏症少年遭受... 詳全文

名稱: 重點是，我愛你
上映日期：2014-02-07
片長:
演員: 歐亨尼奧德伯茲 (Valentin)、Jessica Lindsey
劇情簡介: 瓦倫廷，是阿卡普爾科玩世不恭的花花公子，直到一名妙齡女子莫名地... 詳全文

...

Figure 8: A dialogue simply asks the user to choose eighteen movie type. The order of option list is as initial belief of test1.

Table 5: Expected Reward

	test 1		**test 2**		**test 3**		**Figure 8**	
	B	**t**	**B**	**t**	**B**	**t**	**B**	**t**
B and t	[0.2968, 0.0662, 0.0662, 0.0662, 0.0662, 0.0662, 0.0402, 0.0402, 0.0402, 0.0402, 0.0402, 0.0402, 0.0402, 0.0244, 0.0244, 0.0244, 0.009, 0.009]	[1] [2, 3, 4, 5, 6] [7, 8, 9, 10, 11] [12, 13, 14, 15] [17, 18]	[0.1989, 0.1989, 0.1207, 0.1207, 0.0731, 0.0731, 0.0443, 0.0443, 0.0269, 0.0269, 0.0163, 0.0163, 0.0099, 0.0099, 0.0060, 0.0060, 0.0036, 0.0036]	[1, 2, 3] [4, 5, 6, 7, 8] [9, 10, 11, 12] [13, 14, 15, 16] [17]	[0.0555, 0.0555, 0.0555, 0.0555, 0.0555, 0.0555, 0.0555, 0.0555, 0.0555, 0.0555, 0.0555, 0.0555, 0.0555, 0.0555, 0.0555, 0.0555, 0.0555, 0.0555]	[4, 5, 6, 7, 8] [9, 10, 11, 12, 13] [14, 15, 16, 17] [1, 2, 3]	1	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]
B_{other}	0.0244		0.0036		0.0555			
Expected Reward	**7.39**		**8.6258**		**3.3847**		-3	

References

[1] A. R. Cassandra, "A survey of pomdp applications," in *Working Notes of AAAI 1998 Fall Symposium on Planning with Partially Observable Markov Decision Processes*, pp. 17–24, 1998.

[2] J. Chu-Carroll, "Mimic: An adaptive mixed initiative spoken dialogue system for information queries," in *Proceedings of the sixth conference on Applied natural language processing*, pp. 97–104. Association for Computational Linguistics, 2000.

[3] F. Jurcicek, B. Thomson, and S. Young, "Reinforcement learning for parameter estimation in statistical spoken dialogue systems," *Computer Speech & Language*, vol. 26, no. 3, pp. 168–192, 2012.

[4] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *International Conference on Machine Learning (ICML'95)*, pp. 362–370, Citeseer, 1995.

[5] Q. Liu and S. Li, "Word similarity computing based on how-net," *Computational Linguistics and Chinese Language Processing*, vol. 7, no. 2, pp. 59–76, 2002.

[6] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 3 edition, 2009.

[7] O. Sigaud and O. Buffet, *Markov Decision Processes in Artificial Intelligence*, John Wiley & Sons, 2010.

[8] M. T. Spaan and N. A. Vlassis, "Perseus: Randomized point-based value iteration for pomdps," *Journal of Artificial Intelligence Research*, vol. 24, pp. 195–220, 2005.

[9] C. H. Wu, J. F. Yen, and G. L. Yan, "Speech act modeling and verification in spoken dialogue systems," Chap. 14, pp. 321, World Scientific Publishing Co., Inc., 2006.

Oscar Li Jen Hsu is a Ph.D student in Institute of Information Systems and Applications, National Tsing Hua University, Taiwan. His current research topic is artificial intelligence including decision making and Chinese natural language processing.

Von-Wun Soo was born in Taiwan. He received his Ph.D degree from the Department of Computer Science, Rutgers University, New Jersey, in 1987. He joined as associated professor in Department of Computer Science of

National Tsing Hua University, Taiwan in 1988. He is currently professor at Institute of Information Systems and Applications and Department of Computer Science of NTHU. His research areas include many fields in artificial intelligence such as machine learning, natural language acquisition, intelligent agents, expert systems, ontological engineering and bioinformatics.

A Multi-secret Sharing Scheme Based on the CRT and RSA

Xuedong Dong
College of Information Engineering, Dalian University
10th Street, Dalian Economic Technological Development Zone, Dalian, Liaoning 116622, China
(Email: dongxuedong@sina.com)
(Received Sept. 10, 2013; revised and accepted July 23, 2014)

Abstract

In this paper, we propose to enhance threshold secret sharing schemes based on the Chinese remainder theorem (CRT) by incorporating the well-known RSA Cryptosystem.In the proposed scheme, participants select their secret shadows by themselves. Also, a secure channel among the dealer and participants is no longer needed. In addition,each participant can check whether another participant provides the true secret shadow or not.Furthermore, it allows to reconstruct several secrets parallelly. The scheme is based on the RSA cryptosystem and intractability of the Discrete Logarithm.

Keywords: Chinese remainder theorem; Discrete logarithm; RSA; Threshold secret sharing.

1 Introduction

In a (t, n)-threshold secret sharing scheme, a secret is shared among n participants in such a way that any t (or more) of them can reconstruct the secret while a group of $t-1$ or fewer can not obtain any information. The idea of a secret sharing scheme was first introduced independently by Shamir [16] and Blakley [3], both in 1979. A threshold secret sharing scheme has many practical applications, such as opening a bank vault, launching a nuclear, or authenticating an electronic funds transfer. There are several threshold secret sharing schemes based on the Chinese remainder theorem(CRT) [1, 2, 6, 7, 9, 11, 12, 13, 14, 17]. In these secret sharing schemes there are several common drawbacks as follows [18]:

1) Only one secret can be shared during one secret sharing process;
2) Once the secret has been reconstructed, it is required that the dealer redistributes a fresh shadow over a security channel to every participant;
3) A dishonest dealer may distribute a fake shadow to a certain participant, and then that participant would subsequently never obtain the true secret;
4) A malicious participant may provide a fake share to other participants, which may make the malicious participant the only one who gets to reconstruct the true secret.

In this paper, we propose to enhance threshold secret sharing schemes based on the CRT by incorporating the well-known RSA Cryptosystem invented by Rivest, Shamir, and Adleman [15]. The proposed threshold secret sharing scheme has the following features.

1) Participants select their secret shadows by themselves;
2) A secure channel among the dealer and participants is no longer needed;
3) Each participant can check whether another participant provides the true secret shadow or not;
4) It allows to reconstruct several secrets parallelly.

The scheme is based on the RSA cryptosystem and intractability of the Discrete Logarithm.

The rest of this paper is organized as follows. In Section 2, we give some preliminaries about the CRT. A brief review is given in Section 3, about threshold secret sharing schemes based on the CRT. In Section 4, we propose a new threshold secret sharing scheme based on the CRT by incorporating the well-known RSA Cryptosystem. Section 5 gives the analysis of the proposed scheme. Finally, concluding remarks are given in Section 6.

2 Preliminaries

Several versions of the CRT have been proposed. The next one is called the general CRT [4, 10].

Theorem 1. *Let $k \geq 2, p_1 \geq 2, \cdots, p_k \geq 2$, and $b_1, \cdots, b_k \in Z$. The system of equations*

$$\begin{cases} x \equiv b_1 (\text{mod } p_1) \\ x \equiv b_2 (\text{mod } p_2) \\ \quad\quad \vdots \\ x \equiv b_k (\text{mod } p_k) \end{cases}$$

has solutions in Z if and only if $b_i \equiv b_j (\text{mod } (p_i, p_j)$, for all $1 \leq i, j \leq k$. Moreover, if the above system of equations has solutions in Z, then it has a unique solution in $Z_{[}p_1, \cdots, p_k]$, where $[p_1, \cdots, p_k]$ is the least common multiple of $p_1, \cdots, p_k$.

When $(p_i, p_j) = 1$, for all $1 \leq i, j \leq k$, one gets the standard version of the CRT. Garner [5] has found an efficient algorithm for this case and Fraenkel [4] has extended it to the general case.

3 Brief Reviews

3.1 Review of Mignotte's Threshold Secret Sharing Scheme

Mignotte's threshold secret sharing scheme [9] uses some special sequences of integers, referred to as the Mignotte sequences. Let n be a positive integer, $n \geq 2$, and $2 \leq t \leq n$. An (t, n)-Mignotte sequence is a sequence of pairwise co-prime positive integers $p_1 < p_2 < \cdots < p_n$ such that $\prod_{i=0}^{t-2} p_{n-i} < \prod_{i=1}^{t} p_i$.

Given a publicly known (t, n)-Mignotte sequence, the scheme works as follows:

1) The secret S is chosen as a random integer such that $\prod_{i=0}^{t-2} p_{n-i} < S < \prod_{i=1}^{t} p_i$,

2) The shares I_i are chosen as $I_i \equiv S(\text{mod} p_i)$, for all $1 \leq i \leq n$;

3) Given t distinct shares $I_{i_1}, \cdots, I_{i_t}$, the secret S is recovered, using the CRT, as the unique solution modulo $p_{i_1} \cdots p_{i_t}$ of the system.

$$\begin{cases} x \equiv I_{i_1} (\text{mod } p_{i_1}) \\ x \equiv I_{i_2} (\text{mod } p_{i_2}) \\ \quad\quad \vdots \\ x \equiv I_{i_t} (\text{mod } p_{i_t}). \end{cases}$$

3.2 Review of Asmuth-Bloom's Threshold Secret Sharing Scheme

This scheme, proposed by Asmuth and Bloom in [1], also uses some special sequences of integers. More exactly, a sequence of pairwise co-prime positive integers $p_0, p_1 < p_2 < \cdots < p_n$ is chosen such that $p_0 \prod_{i=0}^{t-2} p_{n-i} < \prod_{i=1}^{t} p_i$.

Given a publicly known Asmuth-Bloom sequence, the scheme works as follows:

1) The secret S is chosen as a random element of the set Z_{p_0};

2) The shares I_i are chosen as $I_i = (S + \gamma p_0)(\text{mod } p_i)$, for all $1 \leq i \leq n$, where γ is an arbitrary integer such that $(S + \gamma p_0) \in Z_{p_1 \cdots p_t}$;

3) Given t distinct shares $I_{i_1}, \cdots, I_{i_t}$, the secret S is recovered as $S = x_0 (\text{mod } p_0)$, where x_0 is obtained,using the CRT, as the unique solution modulo $p_{i_1} \cdots p_{i_t}$ of the system

$$\begin{cases} x \equiv I_{i_1} (\text{mod } p_{i_1}) \\ x \equiv I_{i_2} (\text{mod } p_{i_2}) \\ \quad\quad \vdots \\ x \equiv I_{i_t} (\text{mod } p_{i_t}). \end{cases}$$

4 Proposed Scheme

Let $\{P_1, P_2, \cdots, P_n\}$ be a set of participants and D the dealer of the scheme.The scheme needs a bulletin board.Only the dealer D can change and update the information on the bulletin board and other persons can read and download the information from the bulletin board.

4.1 Initialization phase

1) The dealer D chooses two strong primes $p = 2p^{'}+1$ and $q = 2q^{'}+1$, where $p^{'}$ and $q^{'}$ are also primes. Both p and q should be so safe that anybody can't factor $N = pq$ efficiently. Then the dealer chooses an integer g such that $1 < g < N, (g, N) = 1$ and $(g \pm 1, N) = 1$. Then the order of g is equal to $p^{'}q^{'}$ or $2p^{'}q^{'}$ [2]. D publishes system information $[g, N]$ on the bulletin board and keeps p and q in secret.Each participant P_j randomly chooses a secret integer s_j from $[2, N]$ as her/his own secret shadow, and computes $R_j = g^{s_j} (\bmod N)$, and then sends R_j to D. D must make sure that R_i and R_k are different when $i \neq k$. If $R_i = R_k$, D asks these participants to choose secret shadows again until $R_1, \cdots, R_n$ are different.

2) D chooses the secret integer e, $1 < e < \phi(N) = (p-1)(q-1)$, such that $(e, \phi(N)) = 1$, computes $R_0 = g^e(\mathrm{mod} N)$ and then uses extended Euclidean algorithm to compute a unique integer h, $1 < h < \phi(N)$, such that $eh \equiv 1(\mathrm{mod}\phi(N))$. D publishes R_0, h on the bulletin board.

3) D chooses positive integers $p_1, \cdots, p_n$ such that $\max_{1\leq i_1<\cdots<i_{t-1}\leq n}([p_{i_1}, \cdots, p_{i_{t-1}}]) < \min_{1\leq i_1<\cdots<i_t\leq n}([p_{i_1}, \cdots, p_{i_t}])$, i.e., the sequence $p_1, \cdots, p_n$ is a generalized Mignotte sequence. Then D publishes the sequence $p_1, \cdots, p_n$ on the bulletin board.

4.2 Divide Secret Phase

Suppose that $S_1, \cdots, S_k$ are k secrets to be shared such that $\max_{1\leq i_1<\cdots<i_{t-1}\leq n}([p_{i_1}, \cdots, p_{i_{t-1}}]) < S_w < \min_{1\leq i_1<\cdots<i_t\leq n}([p_{i_1}, \cdots, p$ where $w = 1, \cdots, k$. The dealer D computes $y_{ij} = S_j(\mathrm{mod} p_i) \oplus {R_i}^e(\mathrm{mod} N)$, where $\oplus$ denotes the XOR operation, i.e., componentwise addition modulo 2. D publishes triples (p_i, R_i, y_{ij}), where $i = 1, \cdots n, j = 1, \cdots k$ on the bulletin board.

4.3 Recover Secret Phase

Without loss of generality, assume that participants $P_1, P_2, \cdots, P_t$ cooperate to reconstruct the secret data S_j.

1) Each participant $P_v, v = 1, 2, \cdots, t$ downloads public information R_0, h, and uses her/his secret shadow s_v to compute ${R_0}^{s_v}(\mathrm{mod} N)$ and then sends it and $R_v = g^{s_v}(\mathrm{mod} N)$ to the designated combiner.

2) After receiving ${R_0}^{s_v}(\mathrm{mod} N)$ and $R_v = g^{s_v}(\mathrm{mod} N)$, the designated combiner computes $({R_0}^{s_v})^h(\mathrm{mod} N)$,and checks whether ${R_0}^{hs_v} \equiv R_v(\bmod N)$ is true or not.If ${R_0}^{hs_v} \not\equiv R_v(\bmod N)$, the designated combiner knows that P_v does not provide her/his true secret shadow s_v.

3) The designated combiner downloads public information (p_i, R_i, y_{ij}) on the bulletin board, where $i = 1, \cdots, t$, and computes $y_{ij} \oplus {R_0}^{s_i}(\bmod N) = S_j(\bmod p_i) \oplus {R_i}^e(\bmod N) \oplus {R_0}^{s_i}(\bmod N) = S_j(\bmod p_i)$,where $i = 1, \cdots, t$.

4) The designated combiner uses the general CRT to solve the system of equations

$$\begin{cases} x \equiv y_{1j} \oplus {R_0}^{s_1}(\mathrm{mod} N)(\mathrm{mod}\ p_1) \\ x \equiv y_{2j} \oplus {R_0}^{s_2}(\mathrm{mod} N)(\mathrm{mod}\ p_2) \\ \vdots \\ x \equiv y_{tj} \oplus {R_0}^{s_t}(\mathrm{mod} N)(\mathrm{mod}\ p_t) \end{cases}$$

and gets the general solutions $S_j + [p_1, \cdots, p_t]u$, where $u \in Z$. The unique nonnegative solution less than $[p_1, \cdots, p_t]$ is the secret data S_j.

5 Analysis of the Scheme

5.1 Verification Analysis

From the Euler Theorem it follows that $g^{\phi(N)} \equiv 1(\mathrm{mod} N)$. If P_v is not a cheater, then ${R_0}^{hs_v} \equiv g^{ehs_v} \equiv g^{s_v} = R_v(\mathrm{mod} N)$ since $eh \equiv 1(\mathrm{mod}\phi(N))$. Otherwise, P_v does not provide her/his true secret shadow.

Remark: If a malicious participant randomly chooses an integer s from the range of 2 to N, then performs the subsequent procedures based on s instead of s_v, she/he can pass the above verification successfully. However, $R_s = g^s(\text{mod}N)$ is not equal to any one of R_i in the public information (p_i, R_i, y_{ij}) on the bulletin board, where $i = 1, \cdots, t$. So, the combiner can identify the cheater.

5.2 Security Analysis

1) Having only $t-1$ distinct shares $y_{i_1 j}, \cdots, y_{i_{t-1} j}$, one can only get that $S_j \equiv x_0(\bmod [p_{i_1}, \cdots, p_{i_{t-1}}])$, where x_0 is the unique solution modulo $[p_{i_1}, \cdots, p_{i_{t-1}}]$ of the resulted system(in this case, $S_j > \max_{1\leq i_1 < \cdots < i_{t-1} \leq n}([p_{i_1}, \cdots, p_{i_{t-1}}]) > x_0$).

2) If system attacker personates the dealer to publish a pseudo secret data, she/he has to get the secret number e. Since $R_0 = g^e(\text{mod}N)$, she/he is faced with the difficulty in solving the discrete logarithm problem. Another method of getting e is to solve the equation $eh \equiv 1(\text{mod}\phi(N))$. This needs factorization N into a product of primes which is also difficult.

3) In the secret reconstruction phase, each participant only provides a public value and does not have to disclose her/his secret shadow.Anyone who wants to get the participant's secret shadow will be faced with the difficulty in solving the discrete logarithm problem. The reuse of the secret shadow is secure.

4) Kima *et al.* [8] proposed new modular exponentiation and CRT recombination algorithms which are secure against all known power and fault attacks.

5.3 Performance Analysis

There are efficient algorithms for modular exponentiation and CRT recombination [4, 5, 8]. The XOR operation is of negligible complexity. What's more, each participant chooses her/his secret shadow by her/himself in the proposed scheme, P_j computes $R_j = g^{s_j}(\text{mod}N)$, this also cuts the computation quantity of D.In addition, the system doesn't need a security channel, which also cuts the cost of the system.Therefore the proposed scheme is efficient and practical.

6 Concluding Remarks

This paper proposes to enhance threshold secret sharing schemes based on the CRT by incorporating the well-known RSA Cryptosystem. In the proposed scheme, participants select their secret shadows by themselves. In addition, each participant can check whether another participant provides the true secret shadow or not. Furthermore, it allows to reconstruct several secrets parallelly. Moreover, a security channel is no needed for the proposed scheme. The property is very practical in the system which is unlikely to have a security channel. The scheme is based on the RSA cryptosystem and intractability of the Discrete Logarithm.

Acknowledgments

This study was supported by the National Nature Science Foundation of China under grant 10171042. The author gratefully acknowledges the three anonymous reviewers for their valuable comments.

References

[1] C. A. Asmuth and J. Bloom, "A modular approach to key safeguarding," *IEEE Transactions on Information Theory*, vol. 29, pp. 208-210, 1983.
[2] G. Ateniese, J. Camenisch, M. Joye and G. Tsudik, "A practical and provably secure coalition resistant group signature scheme," in *Proceedings of CRYPTO'00*, pp. 255-270, Santa Barbara, USA, 2000.
[3] G. Blakley, "Safeguarding Cryptographic Keys," *Proceedings of the National Computer Conference*, AFIPS, vol. 48, pp.313-317, 1979.
[4] A. S. Fraenkel, "New proof of the generalized CRT," *Proceedings of American Mathematical Society*, vol. 14, pp. 790-791, 1963.
[5] H. Garner, "The residue number system," *IRE Transactions on Electronic Computers*, vol.8, pp. 140-147, 1959.
[6] K. Kaya and A. A. Selcuk, "Robust threshold schemes based on the chinese remainder theorem," in *Africacrypt'08*, pp. 94-108, 2008.

[7] K. Kaya and A. A. Selcuk, "A verifiable secret sharing scheme based on the chinese remainder theorem," in *Indocrypt'08*, LNCS 5365, pp. 414-425, Dalian, China, 2008.
[8] S. K. Kima, T. H. Kima, D. G. Hanb, S. Honga, "An efficient CRT-RSA algorithm secure against power and fault attacks, "*Journal of Systems and Software*, vol. 84, no. 10, pp. 1660-1669, 2011.
[9] M. Mignotte, "How to share a secret," in *Proceedings of the Workshop on Cryptography*, Burg Feuerstein, 1982, Lecture Notes in Computer Science, vol. 149, pp. 371-375, 1983.
[10] O. Ore, "The general CRT," *American Mathematical Monthly*, vol. 59, pp. 365-370, 1952.
[11] M. Quisquater, B. Preneel, and J. Vandewalle, "On the security of the threshold scheme based on the Chinese remainder theorem," in *PKC'2002*, LNCS 2274, pp. 199-210, Heidelberg, 2002.
[12] S. Y. V. Rao and C. Bhagvati, "CRT based secured encryption scheme," in *1st International Conference on Recent Advances in Information Technology (RAIT'12)*, pp. 11-13, Dhanbad, 2012.
[13] S. Y. V. Rao and C. Bhagvati, "Multi-secret com- munication scheme," in *ICIET'12*, pp. 201-203, Mumbai, 2012.
[14] S. Y. V. Rao and C. Bhagvati,"CRT based threshold multi secret sharing scheme," *International Journal of Network Security*, vol. 16, no. 3, PP. 194-200, May 2014.
[15] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Commununications of the ACM*, vol. 21, pp. 120-126, 1978.
[16] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, pp. 612-613, 1979.
[17] Q. Shi and W. Du,"Multi-secret sharing scheme CRT based on RSA and remainder theorem," *Computer Engineering* (in Chinese), vol. 37, no. 2, PP. 141-142, Jan. 2011.
[18] J. Zhao, J. Zhang, R. Zhao, "A practical verifiable multi-secret sharing scheme," *Computer Standards & Interfaces*, vol. 29, pp. 138-141, 2007.

Xuedong Dong received his Ph.D degree from Nanyang Technological University in 1999. He is currently a Professor in College of Information Engineering, Dalian University. His research interest includes Cryptography, Coding Theory etc. He has published about 30 research papers in journals and conferences.

Guide for Authors

International Journal of Electronics and Information Engineering

International Journal of Electronics and Information Engineering (IJEIE) will be committed to the timely publication of very high-quality, peer-reviewed, original papers that advance the state-of-the art and applications of Electronics and Information Engineering. Topics will include, but not be limited to, the following: Algorithms, Bioinformatics, Computation Theory, AI, Fuzzy Systems, Embedded Systems, VLSI/EDA, Computer Networks, Information Security, Database, Data Mining, and Information Retrieval, Digital Content, Digital Life, and Human Computer Interaction, Image Processing, Computer Graphics, and Multimedia Technologies, Information Literacy, e-Learning, and Social Media, Mobile Computing, Wireless Communications, and Vehicular Technologies, Parallel, Peer-to-peer, Distributed, and Cloud Computing, Semiconductor, Software Engineering and Programming Languages, Telecommunication, etc.

1. Submission Procedure

Authors are strongly encouraged to submit their papers electronically by using online manuscript submission at http://ijeie.jalaxy.com.tw/.

2. General

Articles must be written in good English. Submission of an article implies that the work described has not been published previously, that it is not under consideration for publication elsewhere. It will not be published elsewhere in the same form, in English or in any other language, without the written consent of the Publisher.

2.1 Length Limitation:

All papers should be concisely written and be no longer than 30 double-spaced pages (12-point font, approximately 26 lines/page) including figures.

2.2 Title page

Title page should contain the article title, author(s) names and affiliations, address, an abstract not exceeding 100 words, and a list of three to five keywords.

2.3 Corresponding author

Clearly indicate who is willing to handle correspondence at all stages of refereeing and publication. Ensure that telephone and fax numbers (with country and area code) are provided in addition to the e-mail address and the complete postal address.

2.4 References

References should be listed alphabetically, in the same way as follows:

For a paper in a journal: M. S. Hwang, C. C. Chang, and K. F. Hwang, ``An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445--446, 2002.

For a book: Dorothy E. R. Denning, *Cryptography and Data Security*. Massachusetts: Addison-Wesley, 1982.

For a paper in a proceeding: M. S. Hwang, C. C. Lee, and Y. L. Tang, ``Two simple batch verifying multiple digital signatures," in *The Third International Conference on Information and Communication Security (ICICS2001)*, pp. 13--16, Xian, China, 2001.

In text, references should be indicated by [number].

2.5 Author benefits

No page charge is made.

Subscription Information

Individual subscriptions to IJEIE are available at the annual rate of US$ 200.00 or NT 6,000 (Taiwan). The rate is US$1000.00 or NT 30,000 (Taiwan) for institutional subscriptions. Price includes surface postage, packing and handling charges worldwide. Please make your payment payable to "Jalaxy Technique Co., LTD." For detailed information, please refer to http://ijeie.jalaxy.com.tw or Email to ijeieoffice@gmail.com.