ANALYSIS OF PRIVACY AND NON-REPUDIATION ON PAY-TV SYSTEMS

Ronggong Song and Michael R. Lyu
Computer Science & Engineering Department
Chinese University of Hong Kong
E-mail: rg_song@yahoo.com, lyu@cse.cuhk.edu.hk

Abstract

Lee-Chang-Lin-Hwang in 2000 proposed a set of protocols for Pay-TV systems in order to secure subscriber's privacy and build a fair Pay-TV system. However, we have found that an attacker can easily get other subscriber's privacy in watching TV-programs. We analyze the reason and discuss the possible amendments. Moreover, we expose a weakness on non-repudiation and suggest an improvement to support non-repudiation.

1 Introduction

A Pay-TV system is a commercial TV system, like digital cable TV systems or digital broadcasting systems [1,2,3], which provides TV programs to its subscribers and charges them a subscription fee. A mechanism called a Conditional Access System (*CAS*) is employed on Pay-TV systems to permit only subscribers to watch their designated TV programs. A preferred *CAS* should have the following functions: selectivity, adaptation, suspension, privacy and non-repudiation.

As the technologies to collect and analyze personal information advance, privacy is becoming more and more precious in modern society. In addition, non-repudiation service [4,5] is required to protect the transacting parties from any false denial of payment or service. Thus, keeping subscribers viewing preference secret and reducing possible disputes on Pay-TV systems have become interesting and important.

In order to secure subscriber's privacy and build a fair Pay-TV system, Lee-Chang-Lin-Hwang [3] proposed a set of protocols for Pay-TV systems. In the set of protocols, a registration protocol is used to register a subscriber's identity and deal with his/her premium channels, an adaptation protocol is used to re-select premium channels, and a suspension protocol is used to stop all premium channel services. However, we have found that an attacker can easily get the same TV-programs as the attacked subscriber's by a passive attack. The reason is that there is no any bond between the channel list and the subscriber's identity. Everyone can replay this channel list for his/her own. However, the attacked subscriber and the System Administrator (*SA*) couldn't find it. We then discuss some possible amendments.

Furthermore, the protocol actually doesn't deal with non-repudiation, because it doesn't ensure that the *SA* distributes the right channels to the subscriber before the *SA* gets the subscriber's signature. Thus, an attacker can create a new channel list with the same price, instead of the subscriber's channel list. It is unfair that the *SA* charges the subscriber the subscription fee only by the subscriber's signature because the subscriber actually doesn't get his channels.

The rest of the paper is organized as follows. Lee-Chang-Lin-Hwang Pay-TV protocols are briefly reviewed in the next section. In Section 3, the privacy of the protocols is discussed and the amendments are proposed. In Section 4, the non-repudiation of the protocols is analyzed and an improvement to support the non-repudiation is suggested. In Section 5, the security analysis of the improvement is devoted.

2 Review of Lee-Chang-Lin-Hwang Pay-TV protocol

2.1 Terminology and notations

In a *CAS* for a Pay-TV system, the *SA* provides a range of basic service channels and premium channels. Premium channels consist of Pay-Per-Channel (*PPC*) channels and Pay-Per-View (*PPV*) channels. *PPC* reception fees are

Contributed Paper
Manuscript received May 24, 2001

determined according to time units, while *PPV* reception fees are determined by programs. A Charging Time Period (*CTP*) is a specified period time over which a reception fee is collected.

Notations used in the paper are defined as follows.

- ID_U : user *U*'s identity
- SK_U : user *U*'s secret key
- PK_U : user *U*'s public key
- K_S : a shared key
- T_U : user *U*'s local current time
- R_U : a random number generated by *U*
- $H(.)$: a one-way hash function
- MPK_U : user *U*'s master private key
- AK_j : authorization key of channel *j*
- CW_j : control word of channel *j*
- $E_{PK_U}(M)$: message *M* encrypted with user *U*'s public key
- $S_{SK_U}(M)$: message *M* signed with user *U*'s secret key
- $E_{K_S}(M)$: message *M* encrypted with the shared key
- *Channels* : selected premium channels
- *REGISTER* : registration request command
- *ADAPT* : adaptation request command
- *SUSPEND* : suspension request command
- $ReceptionFee_U$: the price asked of User *U* for the additional selected channels (in the rest of the current *CTP*) or program
- $BalanceFee_U$: the remaining fee of previous registered channels in the rest of the current *CTP*
- $RefundFee_U$: the fee that the *SA* will refund to user *U* for the suspended selected channels in the rest of the current *CTP*
- $DiffFee_U$: the fee that subscriber *U* will pay to the *SA* for the selected channels in the rest of the current *CTP*
- $A \rightarrow B{:}X$: user *A* sends message *X* to user *B*

2.2 Lee-Chang-Lin-Hwang Pay-TV protocols

The *PPC* scheme consists of a registration protocol, an adaptation protocol and a suspension protocol. The *PPV* scheme consists of only a registration protocol and a subscription protocol.

In the *PPC* scheme, when a subscriber, say *U*, wishes to register for the Pay-TV system, the *U* and the *SA* execute the registration protocol as follows.

1. $U \rightarrow SA : S_{SK_U}(ID_{SA}, ID_U, REGISTER\ , T_U)$
2. $U \rightarrow SA : S_{SK_U}(ID_{SA}, ID_U, ReceptionF\ ee_U, T_U)$

   $E_{PK_{SA}}(Channels, R_U)$
3. $SA \rightarrow U : S_{SK_{SA}}(ID_U, H(Channels, R_U), T_{SA})$

When the *U* wants to change his/her registered premium channels, the *U* re-selects his/her favorite channels, Channels', and counts the corresponding fee, $ReceptionFee'_U$. Then the *U* calculates the value:

$$Amount = ReceptionFee'_U - BalanceFee_U.$$

If $Amount > 0$, the *U* and the *SA* execute the adaptation protocol as follows, where $DiffFee_U = Amount$.

1. $U \rightarrow SA : S_{SK_U}(ID_{SA}, ID_U, ADAPT, DiffFee_U, T_U)$

   $E_{PK_{SA}}(Channels', R'_U)$
2. $SA \rightarrow U : S_{SK_{SA}}(ID_U, H(Channels', R'_U), T'_{SA})$

If $Amount <= 0$, the *U* and the *SA* execute the adaptation protocol as follows.

1. $U \rightarrow SA : S_{SK_U}(ID_{SA}, ID_U, ADAPT, T_U)$

   $E_{PK_{SA}}(Channels', R'_U)$
2. $SA \rightarrow U: S_{SK_{SA}}(ID_U, RefundFee_U, H(Channels', R'_U), T'_{SA})$

When the *U* wants to suspend the *PPC* service, the *U* and the *SA* execute the suspension protocol as follows.

1. $U \rightarrow SA : S_{SK_U}(ID_{SA}, ID_U, SUSPEND, T_U)$
2. $SA \rightarrow U : S_{SK_{SA}}(ID_U, RefundFee_U, T_{SA})$

The registration protocol, adaptation protocol and suspension protocol are all followed by the key distribution procedure to allow access control of premium channels. The key distribution includes two cases: one is for new added channels and the other is for deleted channels. If the *U* adds *m* new channels that their keys are AK_j $(j=1, 2, \ldots, m)$, the *SA* broadcasts these keys to the *U* by the following key distribution procedure.

```
For j=1 to m
    $x_{Uj} = E_{MPK_U}(AK_j)$
    broadcast $x_{Uj}$
```

If the U deletes n channels that their key are AK_j ($j=1, 2, \ldots, n$), the SA renews these keys by AK'_j and broadcasts there keys to all subscribers i registered for the channel j by the following key distribution procedure.

```
For j=1 to n
    Renew $AK'_j$
    For all i who registered j
        $X_{ij} = E_{MPK_i}(AK'_j)$
        broadcast $x_{ij}$
```

3 Protection of privacy

The registration protocol and adaptation protocol of PPC exist a flaw in the protection of subscriber's privacy.

3.1 Attack

Here we use the registration protocol as an instance to illustrate the flaw. Suppose the U is going to trigger the registration protocol with the SA. An attacker A can launch to following passive attack by interception to get U's ID_U and the encrypted channel list message: $E_{PK_{SA}}(Channels, R_U)$.

1. $U \to A : S_{SK_U}(ID_{SA}, ID_U, REGISTER, T_U)$

1′. $A \to SA : S_{SK_U}(ID_{SA}, ID_U, REGISTER, T_U)$

2. $U \to A : S_{SK_U}(ID_{SA}, ID_U, ReceptionFee_U, T_U)$
$$E_{PK_{SA}}(Channels, R_U)$$

2′. $A \to SA : S_{SK_U}(ID_{SA}, ID_U, ReceptionFee_U, T_U)$
$$E_{PK_{SA}}(Channels, R_U)$$

3. $SA \to U : S_{SK_U}(ID_U, H(Channels, R_U), T_{SA})$

Then the A can use the encrypted channel list message to subscribe the same channels as the $U's$ as follows.

1. $A \to SA : S_{SK_A}(ID_{SA}, ID_A, REGISTER, T_A)$

2. $A \to SA : S_{SK_A}(ID_{SA}, ID_A, ReceptionFee_A, T_A)$
$$E_{PK_{SA}}(Channels, R_U)$$

3. $SA \to A : S_{SK_{SA}}(ID_A, H(Channels, R_U), T_{SA})$

Then the SA broadcasts the $E_{MPK_A}(AK_j)$. After getting the message $E_{MPK_A}(AK_j)$, the A can find the channels that the U subscribes.

The same attack can be used in the adaptation protocol. The reason is that there is no any bond between the channel list message $E_{PK_{SA}}(Channels, R_U)$ and the subscriber's identity ID_U so that everyone can use the channel list message for his/her own. However, the U and the SA couldn't find it.

3.2 Amendments

Some possible amendments to avoid the above attack are to build a bond between the channel list and the subscriber's identity. There are two kinds of bond cases: one is for directly bond that the subscriber's identity ID_U and the time T_U are added to the channel list as follows,

2. $U \to SA : S_{SK_U}(ID_{SA}, ID_U, ReceptionFee_U, T_U)$
$$E_{PK_{SA}}(ID_U, Channels, R_U, T_U)$$

The other is for indirectly bond that the random R_U is added to the signature message $S_{SK_U}(ID_{SA}, ID_U, ReceptionFee_U, T_U)$ and isn't shown in a plaintext. In the real world, the signature is for the hashing result of the text, not directly for the text itself, like [3], because of high computing complex for signature. Then an amendment for this kind of bond is as follows, where the plaintexts only include the messages: ID_U, $ReceptionFee_U$, T_U.

2. $U \to SA : ID_U, ReceptionFee_U, T_U,$
$$E_{PK_{SA}}(Channels, R_U, T_U)$$
$$S_{SK_U}(H(ID_{SA}, ID_U, ReceptionFee_U, R_U, T_U))$$

Thus, in the first amendment, if an attacker A wants to replay the U's channel list message for his/her own, the SA can easily find that the ID_U doesn't belong to the A. It is impossible for the attacker to discover the channel list by directly comparing because a random number R_U is embedded in the message. In the second amendment, the SA also can easily verify that the signature is not validation because the attacker doesn't know the R_U so that it is difficult for the attacker creating the hashing value containing his/her own identity ID_A and the random number R_U, i.e. $H(ID_{SA}, ID_A, ReceptionFee_U, R_U, T_U)$.

In addition, although there exists a bond between the channel list and the subscriber's identity, the channel list isn't contained in the subscriber's signature and can be forged by the *SA*. Like the original protocol, the *SA* has no ability to prove the subscriber's channel list to someone else. Thus, the privacy property with this protocol is ensured.

4 Support for non-reputation

4.1 Discussion

In Lee-Chang-Lin-Hwang protocols, since there is no any bond between the channel list and the subscriber's identity, it derives another problem: an attacker *A* can launch to following active attack by interception the channel list and create a new channel list with same price, instead of the old channel list as follows.

2. $U \rightarrow A : S_{SK_U}(ID_{SA}, ID_U, ReceptionFee_U, T_U)$

$$E_{PK_{SA}}(Channels, R_U)$$

2'. $A \rightarrow SA : S_{SK_U}(ID_{SA}, ID_U, ReceptionFee_U, T_U)$

$$E_{PK_{SA}}(Channels', R'_U)$$

Thus, although the *SA* hold the subscriber's signature, the subscriber actually doesn't get the channels that he/she selects. It is unfair that the *SA* charges the subscriber the subscription fee by this signature. After completing the protocol, if the subscriber claims that he/she doesn't get the right channels and refuses to pay the subscription fee, the protocol can't provide evidence to enable the disputes resolution.

Furthermore, this problem isn't yet solved even if the protocols are revised like Section 3.2. The reason is that the subscriber doesn't get any information about the channel list that the *SA* distributes to him/her before he/she signs the signature. So after completing the registration protocol, the *SA* can't be sure that the subscriber agrees with the channels that the *SA* distributes.

4.2 Improvement

For the above reason, an improvement protocol is proposed as follows.

1. $U \rightarrow SA : ID_U, REGISTER, T_U$

$$S_{SK_U}(H(ID_{SA}, ID_U, REGISTER, T_U))$$

2. $U \rightarrow SA : E_{PK_{SA}}(K_S), E_{K_S}(ID_U, Channels, R_U, T_U)$
3. $SA \rightarrow U : T_{SA}, S_{SK_{SA}}(H(ID_U, Channels, R_U, T_{SA}))$
4. $U \rightarrow SA : ID_U, ReceptionFee_U, T_U,$

$$S_{SK_U}(H(ID_{SA}, ID_U, ReceptionFee_U, R_U, T_U))$$

The repaired protocol adds the fourth step, but its computing complex is the same as the original protocol. The adaptation protocol can be revised as the above.

According to the repaired protocol, after the fourth step, the *SA* believes that the subscriber agrees with the channels that the *SA* distributes. So after completing the fourth step, the *SA* can broadcast the keys to the subscriber and charge the subscriber the subscription fee.

5 Security analysis

We shall analyze whether the repaired protocol meets the requirements of privacy and non-repudiation.

In the repaired registration protocol for *PPC*, a subscriber's identity ID_U, a random R_U and a time T_U are added to the channel list message. If an attacker wants to replay the channel list message later, the *SA* can find it in Step 2 by T_U. If an attacker wants to replay the channel list message for his/her channel list message, the *SA* can find it in Step 2 by ID_U. If an attacker wants to create a new channel list message instead of the subscriber's channel list message, the subscriber can find it in Step 3 by the *SA*'s signature. And it is impossible for an attacker trying to get the channel list message by comparing because the attacker doesn't know the random R_U. Thus, an attacker can't find the subscriber's channel list however the attacker uses passive or active attack method. In addition, since the subscriber doesn't sign the channel list, the *SA* can't prove to other person that the channel list is made by the subscriber or by himself/herself.

For a Pay-TV system to be considered fair, it must provide non-repudiation services to subscribers and the *SA*. Once the *SA* receives a subscriber's signature in Step 4, the *SA* believes that he/she gives the right channels to the subscriber and that the subscriber agrees with the channels, because the subscriber does the signature only after he/she agrees the channels that the *SA* distributes. Thus, if the *SA* gets the subscriber's signature in Step 4, the subscriber must pay the channel fee. Once the subscriber receives the *SA*'s signature in Step 3, the subscriber can check whether the *SA* gives him/her the right channels.

6 Conclusion

We have shown that Lee-Chang-Lin-Hwang Pay-TV protocols don't provide the protection of privacy and non-repudiation. As a result, an attacker can easily find the subscriber's channel list by subscribing the same channels. Furthermore, we investigate some weakness on non-repudiation in the protocol. We analyze the reasons and propose an improvement scheme to withstand these attacks. Finally, the privacy and non-repudiation of the repaired protocol are examined.

Acknowledgements

We would like to thank the Research Grants Council of the Hong Kong Special Administrative Region. The work described in this paper was substantially supported by a grant from them (Project No. CUHK4193/00E). We are also grateful to the anonymous referees of this paper for their helpful comments.

Reference

[1] T.K.Kim, W.Sohn and S.H.Noh, "The Design and Implementation of TSBB for Koreasat D-DBS (Digital Direct Broadcasting Satellite) System", IEEE Trans. On Consumer Electronics, Vol.43, No.3, pp.330-336, Aug.1997.

[2] F.K.Tu, C.S.Laih and H.H.Tung, "On Key Distribution Management for Conditional Access System on Pay-TV System", IEEE Trans. On Consumer Electronics, Vol.45, No.1, pp.151-158, Feb.1999.

[3] N.Y.Lee, C.C.Chang, C.L.Lin and T.Hwang, "Privacy and Non-repudiation on Pay-TV Systems", IEEE Trans. On Consumer Electronics, Vol.46, No.1, pp.20-26, Feb. 2000.

[4] J.Zhou and D.Gollmann, "Evidence and Non-repudiation", Journal of Network and Computer Application, 20(3), pp.267-281, 1997.

[5] J.Zhou and K.Y.Lam, "Security Digital Signature for Non-repudiation", Computer Communications, (22), pp.710-716, 1999.

Biography

Ronggong Song received his B.Sc degree in mathematics in 1992, M.Sc degree in computer science in 1996, Ph.D. in network security from Beijing University of Posts and Telecommunications in 1999. He had employed as Network Planning Engineer at Telecommunication Planning Research Institute of MII, P.R.China, and Postdoctoral Fellow at University of Ottawa, Canada. His research interests are network security, e-commerce, IP mobility and QoS.

Michael R. Lyu received his B.S. in Electrical Engineering from National Taiwan University in 1981, his M.S. in Computer Engineering from University of California, Santa Barbara, in 1985, and his Ph.D. in Computer Science from University of California, Los Angeles, in 1988. He had been employed as a Technical Staff Member at the Jet Propulsion Laboratory, Assistant Professor at University of Iowa, Member of the Technical Staff of the Bell Communications Research (Bellcore) and research Member of the Technical Staff at Bell Labs. He is currently an Associate Professor at the Chinese University of Hong Kong. His research interests include software reliability engineering, distributed systems, fault-tolerant computing, wireless communication networks, Web technologies, digital library, and E-commerce systems.