Novel Secure Communication Protocol for Conditional Access System

Liu Yongliang, Xiaolin Yang, Hongxun Yao, and Wen Gao
(Corresponding author: Liu Yongliang)

Department of Computer Science and Engineering, Harbin Institute of Technology Harbin No.8, Xue Qing Rd., Hai Dian District, Beijing, 100085, China (Email: liuyongliangs@hotmail.com)
(Received Nov. 29, 2005; revised and accepted Dec. 31, 2005)

Abstract

A protocol for secure communication between set-top box and smart card in conditional access system is proposed. The proposed protocol uses the Schnorr identification scheme to achieve the authentication of smart card to set-top box and uses an asymmetric cryptosystem to achieve the authentication of set-top box to smart card. Both security and performance of the proposed protocol are analyzed and a comparison between the proposed protocol and a previous protocol is provided. The result shows that the protocol is more secure at the cost of a little more computation spending and very applies to smart card with limited processing power. Moreover, the protocol makes it possible that various conditional access systems use the same set-top box because it is not necessary for set-top box to store any secret proprietary data of conditional access system in advance in the protocol.

Keywords: Conditional access system, mutual authentication, session key, set-top box, smart card

1 Introduction

Pay TV has been a physical add on to existing free-to-air TV service infrastructure financed by traditional sources of income such as advertisements, and taxes. It is a discretionary expense and being promoted as offering greater program choice than ever before. Conditional access system (CAS) is an essential component of Pay TV system [3, 4, 11]. It is responsible for ensuring that television programs are accessible only to those customers who have satisfied clearly specified conditions, usually payment related. Conditional access system is mainly composed of two parts: the head-end part and the reception-end part.

1.1 The Head-end

At the head-end, the digital content (including video, audio and data), which the operator wishes to restrict access, is scrambled by the control word (CW) derived from a constantly changing pseudo-random binary sequence generator. The control word also needs to be protected: the control word is encrypted with a service key (SK). The encrypted control word is then packaged into so-called entitlement control message (ECM). Further, the service key is encrypted with the individual key (IK) supplied by the subscriber management system (SMS) and is then packaged with entitlement data into entitlement management message (EMM). Finally, the scrambled content, entitlement control message, and entitlement management message are together broadcasted in the same channel. The process is depicted in Figure 1.

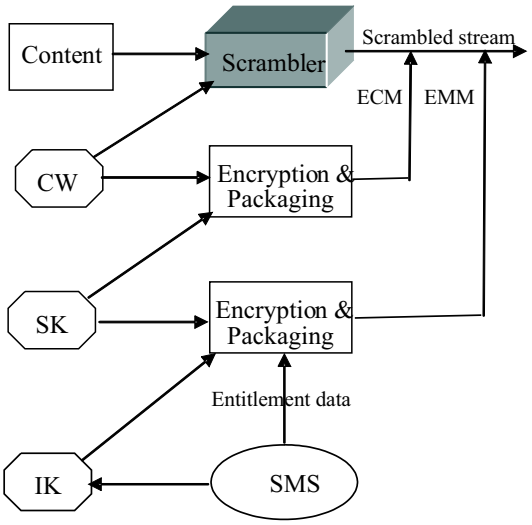


Figure 1: The general description of the head-end

1.2 The Reception-end

At the reception-end, the set-top box (STB) filters entitlement management message and entitlement control message according to the parameters provided by the smart card (SC) and then forwards these messages to smart card. Smart card decrypts entitlement manage-

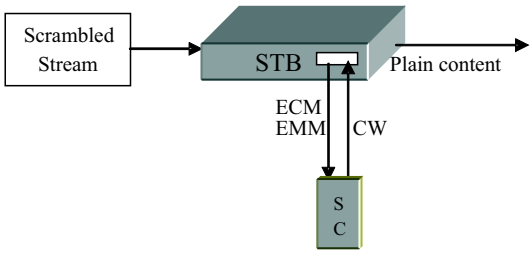


Figure 2: The general description of the reception end

ment message using individual key stored in smart card to get service key and the entitlement data. After having passed the verification of the access entitlement, smart card uses the service key to decrypt the encrypted control word and returns the control word towards set-top box so that set-top box will be allowed to descramble the scrambled content. The process is illustrated in Figure 2.

1.3 Security Threat

The control word is very vulnerable to the link between smart card and set-top box. If smart card transfers the control word in the plaintext form to set-top box, the attacker, instead of being forced to compromise smart card, can obtain the control word by monitoring the interface between smart card and set-top box. Further, the attacker can distribute the control word through radio means or through the Internet to the unauthorized users so that they can enjoy the content freely. This is an indeed serious threat to the security of the conditional access system. Thus, the control word must be protected from this attack. On the other hand, the absence of mutual authentication mechanism[1]. would allow a fake set-top box, for example a computer with hacking smart card reader, to challenge smart card or a pirated smart card to be used on set-top box to access the protected content. So, smart card and set-top box have to authenticate each other to guarantee the system security.

1.4 Related Work

In order to resolve the mentioned above security problem, Jiang et al. proposed a key exchange protocol based on the Schnor's digital signature scheme and one-way hash function [5, 10]. They understood their protocol is dynamic, secure, authenticated, and lower computation. However, we found that there are some deficiencies in their protocol which render their protocol insecure. We will give a brief review of Jiang et al.'s protocol in Section 3.

[1]Mutual authentication can provide two or more communicating parties with some assurance that they know each other's true identity.

1.5 Our Work

In this paper, we present a novel protocol for the secure communication between smart card and set-top box. The proposed protocol uses the Schnorr identification scheme to achieve the authentication of smart card to set-top box and uses an asymmetric cryptosystem to achieve the authentication of set-top box to smart card. Furthermore, we provide the analysis of both security and performance. The rest of this paper is organized as follows: the next Section gives a brief overview of Schnorr identification scheme. In Section 3 Jiang et al.'s protocol is reviewed. In Section 4 we present the proposed protocol and provide methods to improve the performance of the proposed protocol. In Section 5 we provide both the security analysis and performance analysis of the proposed protocol. Then we give the comparison between Jiang et al.'s protocol and the proposed protocol. Finally, a conclusion is given in Section 6.

2 Schnorr Identification Scheme

Schnorr identification scheme can minimize the work to be done by the party with limited resources. And most of computations can be done in preprocessing mode during the idle time of the processor. Thus, Schnorr identification scheme very fits to be implemented on the smart card. Schnorr identification scheme can be divided into three phases: initiation of the trusted authority (TA), registration of the user, and identity authentication.

2.1 Initiation of the Trusted Authority

The TA chooses:

1) Primes p and q such that $q \mid p-1$, $q \geq 2^{140}$, and $p \geq 2^{512}$.

2) $\alpha \in Z_p$, i.e. $\alpha^q = 1 \bmod p$, $\alpha \neq 1$, $Z_p = \{0, 1, \cdots, p-1\}$.

3) A secure one-way hash function $h(\cdot)$ and a secure parameter $t = 72$.

4) Its own public key pk_A and private key sk_A.

The parameters p, q, α, and t, hash function $h(\cdot)$, and public key pk_A are public to all users.

2.2 Registration of the User

Every user chooses a random number s as his private key, $s \in \{1, 2, \cdots, q\}$. The corresponding public key is $v = \alpha^{-s} \bmod p$. When the user comes to the TA for a registration, the TA verifies its identity, generates an identification string I, signs the pair (I, v), and issues the signature to the user.

2.3 Authentication

When the prover P needs to prove its identity to the verifier V, the following steps are performed:

1) Initiation. P sends its identification string I, public key v, and TA's signature on (I, v) to V. V checks the validity of the received message by verifying TA's signature.

2) Preprocessing. P chooses a random number $r \in \{1, \cdots, q-1\}$, computes $x = \alpha^r \bmod p$, and sends x to V.

3) V sends a random number $e \in \{1, \cdots, 2^{t-1}\}$ to p.

4) P computes $y = (r + se) \bmod q$, and sends y to V.

5) Identification test. V checks $x = \alpha^y v^e \bmod p$ and accepts P's proof of identity if and only if equality holds.

3 Review of Jiang et al.'s Protocol

3.1 Registration Phase

When a user applies to subscribe the charge program, the broadcast operator (OP) assigns a smart card with identity ID_c for the user, chooses a random number x_c as private key of smart card, computes $y_c = \alpha^{-x_e} \bmod p$ as public key of smart card. Then OP stores $h(\cdot)$, $E(\cdot)$, ID_C, and ID_s in smart card and stores $h(,)$, $E(\cdot)$, ID_C, ID_S, and x_s in set-top box, where $E(\cdot)$ is a symmetric encryption algorithm, ID_S is the identity of set-top box, and x_s is the secret key of set-top box. Of which ID_S and $h(\cdot)$ are only known to both the smart card and the set-top box.

3.2 Mutual Authentication Phase

1) SC generates two random numbers t and r, computes $T = \alpha^t \bmod p$ and $Y = h(T, ID_C, ID_S)$, and sends ID_C, T, Y and R to STB[2].

2) STB chooses a random number e, $0 \le e \le 2^k$, $k = 72$ (Jiang et al. mistake $k = 72$ for k is 72 bits), computes $M = h(ID_s, r)$, and sends $\{M, e\}$ to SC.

3) SC checks $M = h(ID_s, r)$ true or not. If true, SC accepts STB identity, computes $d = t + ex_c \bmod q$ (Jiang et al. mistake $d = t + ex_c \bmod q$ for $d = t + ex_c \bmod p$), and sends d to STB.

4) STB checks $Y = h(\alpha^d, y_c^e, ID_C, ID_S)$ true or not. If it is true, STB accepts SC identity.

[2]Note that the message sent by smart card in Fig. 3 of Jiang et al.'s paper is X, T, Y, r, and ID_C (it doesn't match the message X, Y, r, and ID_C in the text of Jiang et al.'s paper), where the message X is used for the authentication of the user to STB. Here, we focus on the mutual authentication between set -top box and smart card. So, we have omitted the message X.

3.3 Key Agreement Phase

If mutual authentication is passed successfully for both STB and SC, then they use the following equation to compute a common session key $SK = h(r, e, ID_c, ID_s)$.

3.4 Some Deficiencies in Jiang et al.'s Protocol

We found that there are some deficiencies in Jiang et al.'s protocol which render their protocol insecure. These deficiencies are described as follows:

1) First and the most important, the protocol allows any SC with a fake certificate (with message in the form of the modular exponentiation) pass the authentication to STB. The reason is that the certificate verification required in Schnorr's scheme was missed in their protocol.

2) The authenticity of prime p has not been provided. The Pohlig-Hellman type attack may apply [2, 7].

3) The protocol doesn't provide any key confirmation while it provides entity authentication.

4) The security of the protocol based on the privacy of the hash algorithm is suspicious.

5) The way of authentication of STB to SC based on share secret between SC and STB is not ideal because it is expected in general that STB does not contain any proprietary data in advance for the sake of that same STB can be used by various CAS.

6) The run of the protocol should be initiated by STB rather than by SC.

7) It seems that both x_c and y_c should be stored in SC in registration phase, too.

8) It is wondering how STB obtains the value y_c and α in Step (4) of the mutual authentication phase.

4 Our Protocol

Our protocol has two phases: a preparation phase and a communication phase. These two phases are described respectively as follows.

4.1 The Preparation Phase

The preparation phase involves the trusted authority (TA), which has a pair of public/private keys (pk_A, sk_A), STB manufacturer, and TV broadcast operator (OP) which also functions as secondary trusted authority.

1) TA generates a distinguishing identification string I_S, a pair of public/private keys (pk_S, sk_S), and the corresponding public key certificate $C_A(I_S) = I_S, pk_S, sig_A(I_S, pk_S)$ for each STB,

where $sig_A(I_S, pk_S)$ denotes TA's signature on the message (I_S, pk_S) with private key sk_A. Then TA safely delivers pk_A and $sk_S, C_A(I_S)$ to the STB manufacturer by a trust carrier or through a secure channel between TA and STB manufacturer.

2) The STB manufacturer places the set of the messages $pk_A, C_A(I_S), sk_S$ into the secure memory of each STB at the stage of producing STB.

3) The TA generates a unique identification string I_O, pair of public/private keys (pk_O, sk_O), and a public-key certificate $C_A(I_O) = I_O, pk_O, sig_A(I_O, pk_O)$for each OP. The TA safely delivers the set of messages $pk_A, C_A(I_O), sk_O$to each OP.

4) The OP chooses the parameters p, q, and α (see Section 2.1). Further, the OP selects a symmetric encryption algorithm $E_k(\cdot)$ such as AES, an asymmetric encryption algorithm $\bar{E}_{pk}(\cdot)$such as RSA, and a secure hash algorithm $h(\cdot)$.

5) For each SC, the OP assigns a unique identification string I_C, generates a random number $u < q$ as private key, computes the public -key $v = \alpha^{-u} \bmod p$, and creates the corresponding public-key certificate $C_O(I_C) = (I_C, v, \alpha, p, sig_O(IC, v, \alpha, p))$.Then, OP stores $pk_A, C_O(I_C), pk_O, u, E_k(\cdot), \bar{E}_{pk}(\cdot)$, and $h(\cdot)$ in each SC.

6) For the first time use, STB downloads the OP's public-key certificate $C_A(I_O)$, the secure hash algorithm $h(\cdot)$, the encryption algorithms $E_k(\cdot)$ and $\bar{E}_{pk}(\cdot)$, and the OP's signatures on both $E_k(\cdot)$ and $\bar{E}_{pk}(\cdot)$[3]. Then, STB use pk_A to verify the certificate $C_A(I_O)$ and uses pk_O contained in $C_A(I_O)$ to verify the OP's signatures on both $E_k(\cdot)$ and $\bar{E}_{pk}(\cdot)$. If the result of all verifications is positive, STB stores pk_O,$E_k(\cdot)$, $\bar{E}_{pk}(\cdot)$ and $h(\cdot)$ into its own memory.

4.2 The Communication Phase

Figure 3 depicts the communication phase. In this phase both STB and SC perform the following operations:

1) STB sends the certificate $C_A(I_S)$ to SC.

2) SC verifies the certificate $C_A(I_S)$ using the TA's public-key pk_A. If the result of the verification is positive, SC generates a random number $a \in \{1, \cdots, q\}$, a random nonce r_1, and a random session key K, computes $b = \alpha^a \bmod p$,encrypts b and r_1 with K to get $E_K(b, r_1)$, encrypts K with STB's public key pk_S to get $\bar{E}_{pk_S}(K)$, and sends $\bar{E}_{pk_S}(K)$, $E_K(b, r_1)$, and its own public-key certificate $C_O(I_C)$ to STB.

3) STB verifies the certificate $C_O(I_C)$ using the public-key pk_O. If the result of the verification is positive, STB decrypts $\bar{E}_{pk_S}(K)$ with its private key sk_S to get the r_1, creates a random nonce r_2 and a random number $m \in \{0, \cdots, 2^t - 1\}$,$t = 72$, encrypts m, r_1, and r_2 with session key K to get $E_K(m, r_1, r_2)$, and sends $E_K(m, r_1, r_2)$ to SC.

4) SC decrypts $E_K(m, r_1, r_2)$ with the session key K to get m, r_1, and r_2, checks r_1 agree with that sent in Step (2), computes $M = h(m, I_C, I_S)$ and $c = (a + uM) \bmod q$, encrypts c and $r2$ with K to get $E_{K(c,r_2)}$, and sends $E_{K(c,r_2)}$ to STB. Note that it is important that both identification information I_C and I_S are contained in the hash function for preventing man-in-the-middle attack [6].

5) Upon receiving the message $E_{K(c,r_2)}$, STB decrypts $E_{K(c,r_2)}$ with K to get c and r_2, checks the number r_2 agrees with that sent in Step (3). Provided the result of the check is positive, STB computes $M = h(m, I_C, I_S)$, checks $b = \alpha^c v^M \bmod p$ and accepts the identification proof of SC if this equation holds.

Note that the session key K is saved in the RAM of both SC and STB, respectively. After a separation or power off, the new session key should be redistributed between SC and STB.

4.3 Methods to Improve the Performance

Numerous refinements can be devised to improve the performance of our protocol. We only explain four methods here.

1) RSA with public exponent 3 can be chosen as public-key encryption/signature algorithm to reduce the computational cost. In this case, the public-key encryption or verifying signature is reduced to two modular multiplications.

2) Both creating the random number a and computing the value b can be done by SC long before the protocol runs, using idle time of the processor [10]. This is particularly suited to SC with limited computation power. Rooij presented the attack to the preprocessing in [8], however, Schneier pointed out that attack is impractical [9].

3) Using digital signature algorithm (DSA) to create the certificate for SC will reduce storage and communication cost.

4) Computing $c = (a + uM) \bmod q$ can be done by precomputing (only once, when the private key u of SC is chosen) an approximation e of u/q [1]. If $0 < u/q - e < 2^{-t-1}$, then $a - q < a + uM - q[[eM]] < a + q$ where [[eM]] denotes the nearest integer to eM. Hence after computing $a + uM - q[[eM]]$, at most one subtraction or addition of q will be required to reduce $c = (a + uM) \bmod q$. The overall improvement from performing the precomputation is to replace the division by q with a multiplication of e and M followed

[3]These data, like audio or video data, can be broadcasted as a service for set -top boxs downloading.

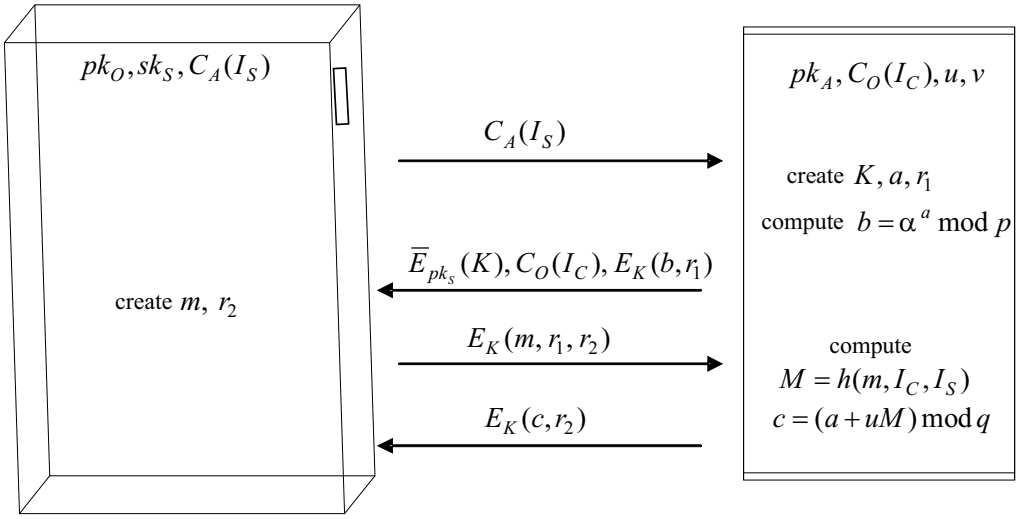


Figure 3: The general description of the communication phase

by multiplication of q, followed by at most two subtractions or additions. Depending on the implementation, this may result in a significant speedup by eliminating the multiple precision divisions.

5 Discussions

5.1 Security Analysis

Consider what assurances our protocol provides to SC. From the point of view of SC, as a result of the key transport, it shares a session key known only to it and the certificate holder that may or may not be STB. By decrypting the message encrypted by SC using the public -key contained in the certificate and returning the message of the random number encrypted with the session key, STB demonstrates to SC that it is very the certificate holder. This gives SC assurance that STB it carried the communication out with is an authorized STB and STB learns the session key. On the other hand, SC authenticates itself to STB by demonstrating knowledge of the private key u. After checking the equation $b \equiv \alpha^c v^m \bmod p$ holds, STB accepts the proof of SC identity. Thus, our protocol provides both entity authentication and explicit key authentication.

The protocol is secure to the *fake SC/STB attack*. Since the fake SC/STB hasn't its own valid public key certificate, the attacker has to forge a certificate or use the intercepted certificate in order to pass the authentication to authorized STB/SC. In the former case, the fake certificate will be detected by STB/SC during the certificate verification. In the latter case, the fake SC/STB can not return the valid message. So, the fake SC/STB can be detected during the SC/STB identification authentication.

The protocol is secure to man-in-the-middle attack. The most possible attack is that the attacker substitutes $E_{pk_S}(K')$ and $E_{K'}(b', r_1')$ for $E_{pk_s}(K)$ and $E_K(b, r_1)$ respectively, K' is known to the attacker. In this case, STB decrypts the $E_{pk_s}(K')$ and then returns $E_{K'}(m, r_1', r_2)$. Having no knowledge of the key K, the attacker could not forge the message to deceive SC. So, SC will detect the abnormity in Step (4) of the communication phase and reject further communication.

The protocol is secure to replay attack. The random nonces r_1, r_2 are used to prevent replay attack. Checking random nonce r_1/r_2 also makes SC/STB sure that received message in Steps (3)/(4) of the communication phase is fresh and does be from the authorized STB/SC.

5.2 Performance Analysis

Here, we concentrate on the modular exponentiation and modular multiplication which require more computational cost. It is well known that SC is with limited calculation power. So, the working load of SC should be as less as possible. In our protocol, SC performs one modular exponentiation and five modular multiplications, assuming that RSA with public exponent 3 was used. Of which the modular exponentiation and one modular multiplication could be precomputed (see Section 4.3). So, SC only needs to perform four modular multiplications online. It is obvious that the computational cost is suitable for SC. On the other hand, STB needs to perform three modular exponentiations and two modular multiplications. These operations can be easily implemented by STB.

5.3 Comparison

The comparison between Jiang et al.'s protocol and our protocol is provided in Table 1.

Table 1: Comparison between Jiang et al.'s protocol and our protocol

			Jiang et al.'s protocol	Our protocol
Reliability of authentication			No. Certificate can be forged due to the absence of certificate verification.	Yes. Authentication is based on certificate, and certificate authentication is provided
Security			Based on privacy of hash function. Pohlig-Hellman type attack may apply	Based on privacy of the keys
PERFO	SC	Total	1 modular exponentiation 1 modular multiplication	1 modular exponentiation 5 modular multiplications
		Online	0	4 modular multiplications
	STB		2 modular exponentiations 1 modular multiplication	3 modular exponentiations 3 modular multiplications
Need to store secret proprietary data (STB)			Yes	No
Compatibility (STB)			No	Possible*

* Our protocol makes it possible that various CAS uses the same STB because of that there is no need for STB to store any CAS secret proprietary data in advance in the protocol.

6 Conclusions

In this paper, a protocol for secure communication between set-top box and smart card in conditional access system is proposed. The protocol uses the Schnorr identification scheme to achieve the authentication of smart card to settop box and uses an asymmetric cryptosystem to achieve the authentication of set-top box to smart card. The protocol minimizes the online computational burden of smart card while provides the same level of security as other protocols. Both security and performance of the protocol are analyzed. The result of the analysis shows that the protocol is robust to the malicious attacks and very applies to smart card with limited processing power. Moreover, the protocol makes it possible that various conditional access systems use the same set-top box because it is not necessary for set-top box to store any secret proprietary data of conditional access system in advance in the protocol.

Acknowledgments

This work was supported by the National Hi-Tech Research and Development Program (863) of China (2004AA119010) and Microsoft Research Asia (TWC-2006-Project-5). The authors are grateful to the anonymous reviewers for valuable comments. While the authors are grateful to Dr. Shaohui Liu and Dr. Bo Wu for helpful discussions.

References

[1] E. F. Brickell and K. S. McCurley, "An interactive identification scheme based on discrete logarithms and factoring," in *Eurocrypt'90*, LNCS 473, pp. 63-71, Springer-Verlag, 1991.

[2] W. Diffie, P. C. Oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes and Cryptography*, vol. 2, no. 2, pp. 107-125, 1992.

[3] EN50094, *Access Control System for the MAC/packet family: EUROCRYPT*, CENELEC, Dec. 1992.

[4] ETSI Technical Report, *Digital Video Broadcasting; Support for Use of Scrambling and Conditional Access (CA) within Digital Broadcasting Systems*, Oct. 1996.

[5] T. Jiang, Y. Hou, and S. Zheng, "Secure communication between Set-top box and smart card in DTV broadcasting," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 3, pp. 882-886, Aug. 2004.

[6] A. Menezes, P. V. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Oct. 1996.

[7] S. C. Pohlig and M. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance," *IEEE Transactions on Information Theory*, vol. 24, no. 1, pp. 106-110, 1978.

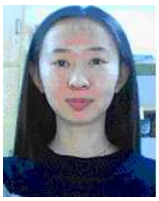
[8] P. D. Rooij, "On Schnorr's preprocessing for digital signature schemes," in *Eurocrypt'93*, LNCS 765, pp. 435-439, Springer-Verlag, Berlin, 1993.

[9] B. Schneier, *Applied Cryptography: Protocols, Algorithm, and Source Code in C*, 2nd Edition, Wiley, 1996.

[10] C. P. Schnorr, "Efficient identification and signatures for smart cards," *Proceedings on Advances in cryptology*, pp. 239-252, Santa Barbara, United States, 1989.

[11] X. Verians, J. M. Boucqueau, *Next Generation Conditional Access System for Satellite Broadcasting*, Final Report, Contract no. 16696/02/NL/US, Published on 12, Jan. 2004.

Liu Yongliang is a Ph.D. candidate at Harbin Institute of Technology. He received his MS degree in Mathematics from Harbin Institute of Technology, China, 2000. His research interests include: digital watermark, cryptographic protocol, and digital right management. He has published 20 scientific papers.



Xiaolin Yang is a teaching assistant at Harbin Institute of Technology. She received her MS degree from Harbin Institute of Technology, China, 2004. Her research interests include: pattern recognition, digital watermark, and digital right management. She has published 5 scientific papers.

Hongxun Yao received her B.S. and M.S. degrees in computer science from Harbin Shipbuilding Engineering Institute, Harbin, China, in 1987 and 1990 respectively, and the Ph.D degree from Harbin Institute of Technology, Harbin, China, in 2003. Her research interests lie in image processing, pattern recognition, multimedia technology and natural human-computer interface and information hiding technology. She has published 4 books and over 60 scientific papers.

WEN GAO received his Ph.D. degree in Computer Science, Harbin Institute of Technology, China, 1988 and Ph.D. in Electronics Engineering, University of Tokyo, Japan, 1991. He was a Research Fellow at Institute of Medical Electronics Engineering, the University of Tokyo, in 1992; a Visiting Professor at Robotics Institute, Carnegia Mellon University, in 1993; a Visiting Professor at MIT AI Lab, from May 1994 to December 1995. Now he is a professor of Institute of Computing Technology, Chinese Academy of Sciences. His researchinterests include pattern recognition and arti.cial intelligence, image understanding, data compression, hand gesture recognition, multimodal interface, and computer vision. He has published 7 books and over 260 scientific papers.