A Trusted Authentication Protocol based on SDIO Smart Card for DRM

Jian Wang, Zhiyong Zhang, Fei Xiang, Lili Zhang, Qingli Chen
Electronics Information Engineering College, Henan University of Science and Technology, Luoyang 471003, China
E-mail: wangjian_migi@sina.com, z.zhang@ieee.org

Abstract

Terminals security vulnerabilities makes DRM researches to focus on trusted computing technology in recent years, however, no efficient and practical trusted authentication protocol is presented, especially with formal proof. To attest the integrity when access to the DRM server, the DRM client need perform mutual authentication and key agreement with the server first, and then use the sharing key to encrypt the integrity values. A novel trusted authentication protocol based on SDIO smart card is presented together with its formal security proof. The proposed protocol is composed of registration phase, login phase, identity authentication and key agreement phase, and integrity attestation phase. In contrast to other corrective schemes through attack resisting analysis and computational cost analysis, the proposed scheme is able to provide greater security and practicality to guarantee the trust attestation for DRM.

Keywords*: DRM, Trusted Authentication, Mutual Authentication And Key Agreement, Formal Proof, Strand Spaces*

1. Introduction

For terminals security vulnerabilities threatens to DRM (Digital Rights Management) system [1,2], DRM researches focus on trusted computing technology in recent years. The thesis [3-5] describe their researches about the DRM system base on trusted computing, however, none of them presents the design of authentication protocols in DRM. Actually, the authentication process of DRM server to client is a remote attestation from the client to the server in trusted computing based DRM systems. TCG specifications [6] define the remote attestation architecture, but never provide detailed protocols. In 2004, Sailer et al. [7] presents a simple remote attestation protocol, which is only a challenge & response process with a nonce. Stumpf et al. [8] showed this protocol can't resist masquerading attack, and presented another robust scheme to improve it. In addition, Tan et al. [9] presented a remote attestation protocol based on TPM, which is named TRAP and used for sensor networks. However, in these papers, there is no formal proof for protocols correctness, but only informal security analysis. This leads to these protocols have no sense for application and popularization. [10] From other angle, Goldman et al. [11], Gasmi et al. [12] and Zhou et al. [13] separately presents their methods for combining trusted computing technology to identity authentication in TLS, in order to report integrity through trusted extended TLS channel. Similarly, Sadeghi et al. [14] described a method for extending IPSec protocol based on trusted computing. Though these protocols successfully realize transmission of integrity report with the aid of traditional security protocols, none of them is a complete trusted authentication protocol being formal proved. Based on our research [15], the current paper not only demonstrates a trusted authentication protocol based on SDIO smart card for DRM system, but also presents its formal proof based on strand spaces model and informal security analysis.

2. Related Literature

2.1 Trusted Authentication in DRM

In a DRM system based on trusted authentication, DRM client should provide identity attestation and integrity report when it request data service from the DRM servers. The trusted authentication protocol is designed to ensure the peer's identity authenticity, and at the same time, complete key agreement for both sides to build a secure channel for integrity report transmission. So, the trusted authentication protocol includes three phases:

doi:10.4156/jdcta.vol6.issue23.26

(1) Building secure channel. Through identity authentication and key agreement, both sides acknowledge each other and obtain sharing key to build secure channel for integrity report. If the authentication is failed, the connection will be terminated.
(2) Integrity attestation. When identity is verified, DRM client transmit the encrypted integrity measurements with the sharing key to the server when it is requested. If the verification is failed, the connection is terminated.
(3) Using secure session. When the integrity attestation is successfully done, the peer can use the built secure channel to exchange information, such as digital content or digital license.

The flow of the trusted authentication between the DRM Server and client is depicted as figure 1.

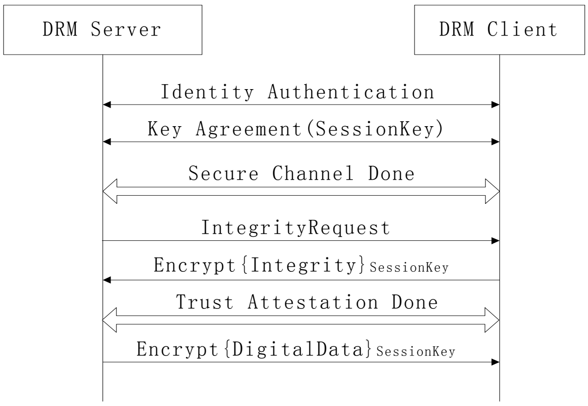

Figure 1. Authentication protocol flow between the DRM Server and client

2.2 Identity Authentication and Key Agreement Protocol Based on SDIO Smart Card

SDIO smart card with its PIN can provide two-factor authentication for terminal users during remote access, which decreases illegal access without authorization and improve networks security significantly. For this reason, the multi-factor authentication technology based on SDIO smart card plays an important role in the identity authentication technology. Since Lamport [16] proposed a remote authentication scheme in 1981, many researchers have proposed new schemes to improve the efficiency and security. Chien et al. [17] proposed an efficient remote mutual authentication scheme using smart card, and claimed that the scheme requires no verification table. However, Hsu [18] showed Chien et al.'s scheme is vulnerable to the parallel session attack. Then, Juang [19] proposed another scheme preserving all the advantages of Chien et al.'s. This scheme is a nonce based protocol, not requiring synchronized clocks, and generating a session key for the user and server in their subsequent communication. However, Shieh-Wang [20] pointed out the weakness of Juang's scheme and then proposed another similar one to improve the weakness. Thereafter, Yoon-Yoo [21] showed Shieh-Wang's scheme does not provide perfect forward secrecy, and is vulnerable to a privileged insider's attack. What's more, we found Shieh-Wang's scheme is still vulnerable to parallel session attack, and can't resist the DoS (Denial of Service) attack using a stolen smart card [15]. Recently, Wang et al. [22] presented cryptanalysis and improvement on other's remote user authentication scheme using smart cards. Yang et al. [23] proposed a new scheme and a generic construction framework for smart-card-based password authentication. Xu et al. [24] presented an improved smart card based password authentication scheme with formal security proof. In 2011, Yoon [25] presented an improved scheme based on the elliptic curve Diffie-Hellman problem and secure one-way hash function in order to isolate previous schemes' security problems. However, there is no formal security proof presented.

3. Trusted Authentication Protocol based on SDIO Smart Card

3.1 Notations

Before presenting the protocol, the notations used in the rest of the paper is listed below.

U*i* : the i th DRM client;
ID*i* : U*i*'s ID;
PW*i* : U*i*'s password;
S : DRM server;
h(.) : Secure one-way hash function;
x : The secret key maintained by the DRM server;
⊕: Logic Exclusive-or operation;
|| : String concatenation operation;
q : A public parameter which is a large prime number;
g : A public parameter which is a primitive element mod q;
N*b* : Nonce value generated by DRM server;
N*a* : Nonce value generated by DRM client;
K*s* : Session key calculated by DRM server;
K*u* : Session key calculated by DRM smart card

3.2 Protocol Describe

The trusted authentication protocol based on SDIO smart card for DRM consists of four phases: the registration phase, the login phase, the identity verification and key agreement phase, and integrity attestation phase.

Registration phase: This phase is invoked whenever a user U*i* initially registers to a remote server S.

(1) U*i* selects his identifier ID*i* and password PW*i,* and submits h(PW*i*) to the remote server over a secure channel.
(2) Upon receiving the registration request, S computes R*i*=h(ID*i*⊕x) ⊕h(PW*i*), C*i*= h(h(ID*i*□x)) □h(PW*i*), and issues U*i* a smart card containing R*i*, C*i* and h(.).

Login phase: this phase occurs when the user wants to login the remote server each time.

(1) U*i* inserts his smart card into the smart card reader of a terminal, and enters his ID*i* and PW*i*.
(2) Smart card firstly check the validity of the password by computing C*i'*= h(a*i*)□h(PW*i*) and checking whether C*i'*=C*i*. If PW*i* is valid, the smart card performs the following steps, or, resists login.
    a) Compute a*i*= R*i*⊕h(PW*i*);
    b) Generate the nonce value N*a* for the DRM client;
    c) Compute DHC$_1$: DHC$_1$ = g^{Na} □a*i*;
    d) Compute MAC$_1$: MAC$_1$ =h(g^{Na}) ;
(3) Send the message (ID*i*, DHC$_1$, MAC$_1$) to S and wait for response from it. If no response is received in time or the response is incorrect, report login failure to the client and stop the session.

Identity verification and key agreement phase: this phase is invoked whenever S receives U*i*'s login request. After receiving the message (ID*i*, DHC$_1$, MAC$_1$) from U*i*, S performs the following steps to assure the integrity of the message, respond to U*i* and challenge U*i*:

(1) Firstly, check the integrity of the message by the following computing: a*i*′ = h(ID*i*⊕*x*)， g^{Na}=DHC$_1$⊕a*i*′ , MAC$_1$′ = h(g^{Na}). If MAC$_1$′ = MAC$_1$, the message is effect. Or, the session is stopped.
(2) Generate the nonce N*b* for S;

(3) Compute DHC$_2$=$g^{Nb} \oplus ai'$, MAC$_2$=h(g^{Na} || g^{Nb})

(4) Send message (DHC$_2$, MAC$_2$) to the client, and wait for its response. If no response is received in time or the response is incorrect, stop the session.

After Ui received the message (DHC$_2$, MAC$_2$) from S, the smart card performs the following steps to authenticate S and respond to S's challenge, and then computes the sharing session key.

(1) Firstly, authenticate S by the following computing: ai=R$i \oplus$ h(PWi), g^{Nb} =DHC$_2 \oplus$ ai, MAC$_2'$=h(g^{Na} || g^{Nb}). If MAC$_2'$= MAC$_2$, the server is trusted and the latter process will be continued. Or , the smart will inform the client that login is failed, and stop the session.

(2) Compute MAC$_3$=h(g^{Nb});

(3) Compute session key Ku: Ku= $(g^{Nb})^{Na}$ mod q;

(4) Send the message (MAC$_3$) to the server.

After receiving the message (MAC$_3$), S performs the following steps to authenticate the client Ui, and obtain the session key.

(1) Computes MAC$_3'$ =h(g^{Nb});

(2) S checks whether MAC$_3'$= MAC$_3$ to verify the client's validity. If equal, S accepts the client's login request, and computes the session key. Or, S refuses the login and stops the session.

(3) Computes session key Ks: Ks= $(g^{Na})^{Nb}$ mod q.

When the identity authentication and key agreement phase is completed, the DRM server and the client can use the session key Ku (=Ks) to encrypt the session data in the latter communications. The message alternation process in the above two phases is illustrated in Figure 2.

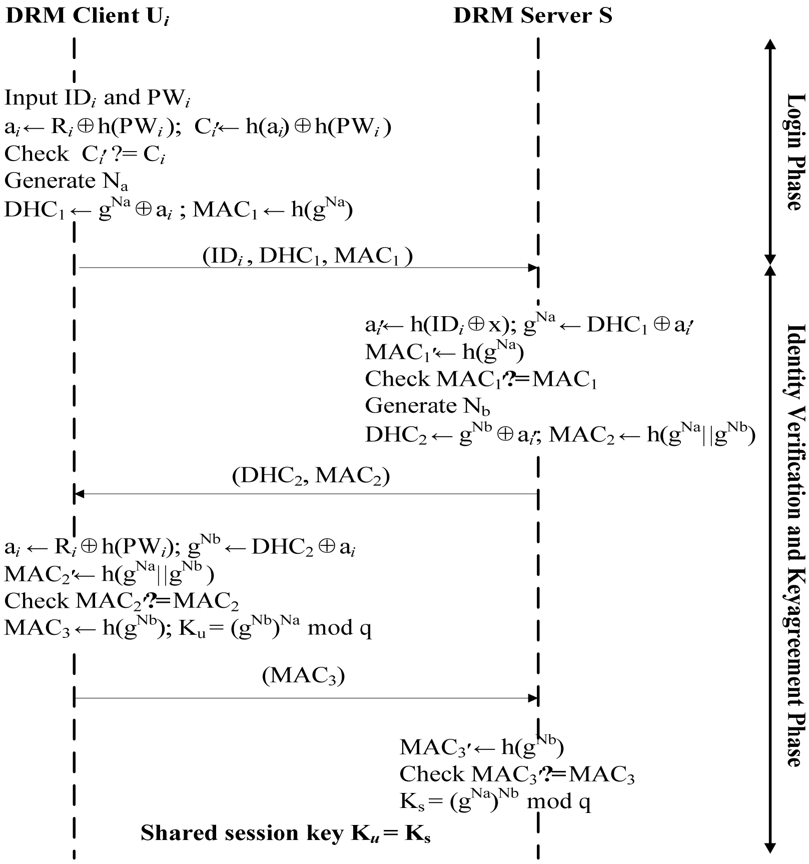
DRM Client U$_i$
DRM Server S
Input ID$_i$ and PW$_i$
a$_i \leftarrow$ R$_i \oplus$h(PW$_i$); C$_i' \leftarrow$ h(a$_i$)$\oplus$h(PW$_i$)
Check C$_i'$?= C$_i$
Generate N$_a$
DHC$_1 \leftarrow$ g$^{Na} \oplus$a$_i$; MAC$_1 \leftarrow$ h(g^{Na})
(ID$_i$, DHC$_1$, MAC$_1$)
a$_i' \leftarrow$ h(ID$_i \oplus$x); g$^{Na} \leftarrow$ DHC$_1 \oplus$a$_i'$
MAC$_1' \leftarrow$ h(g^{Na})
Check MAC$_1'$?=MAC$_1$
Generate N$_b$
DHC$_2 \leftarrow$ g$^{Nb} \oplus$a$_i'$; MAC$_2 \leftarrow$ h(g^{Na}||g^{Nb})
(DHC$_2$, MAC$_2$)
a$_i \leftarrow$ R$_i \oplus$h(PW$_i$); g$^{Nb} \leftarrow$ DHC$_2 \oplus$a$_i$
MAC$_2' \leftarrow$ h(g^{Na}||g^{Nb})
Check MAC$_2'$?=MAC$_2$
MAC$_3 \leftarrow$ h(g^{Nb}); K$_u$= (g^{Nb})^{Na} mod q
(MAC$_3$)
MAC$_3' \leftarrow$ h(g^{Nb})
Check MAC$_3'$?=MAC$_3$
K$_s$= (g^{Na})^{Nb} mod q
Shared session key K$_u$= K$_s$
Login Phase
Identity Verification and Keyagreement Phase

Figure 2. Message alternation process in the above two phases

Integrity attestation phase: this phase occurs when the secure session channel is built. The following process is the remote attestation of TCG. [6]

(1) The server S sends integrity request together with a nonce to the client U*i*. The nonce is sent in order to resist the replay attack.
(2) After receiving the request, the client obtains Quote=Sign{PCRs}PK_{AIK}, which is the PCRs value signed by AIK public key from TPM. Then,
(3) The client computes EM= Encrypt{PCRs, Quote, SML, nonce }$_{Ks}$, and returns EM to the server.
(4) The server decrypt the EM from client with K*s* (=K*u*), and get nonce, PCRs value, Quote and SML.
(5) The server checks nonce to certify the message's freshness, and validates AIK certification to certify the AIK secure keys are valid. Then, the server use AIK's public key to attest the Quote is valid. If all these are trusted, the server will check the PCRs value and SML to make sure whether the client's configuration is trusted.
(6) If the integrity of the client is certified to be right, the server will start the digital content sharing with the client or distributing digital license to it. Or, the server will stop the connection.

The process of this phase is illustrated as the following figure 3.

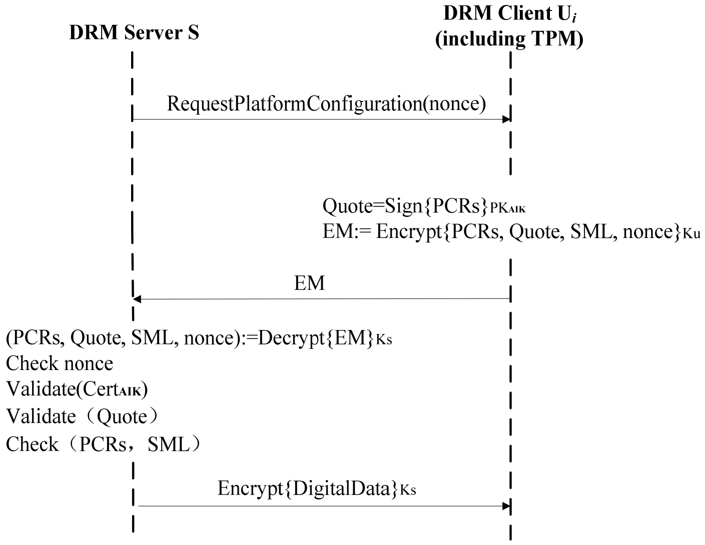

Figure 3. Message alternation process in the integrity attestation phase

By now, the whole trusted authentication protocol for DRM system is completed. Specially, the presented scheme above allows DRM client users to change password freely and securely without remote server's help. Concretely, when U*i* wants to change his password PW*i* with a new one, for example PW*i**, the following steps will be performed:

(1) U*i* inserts his smart card into the smart card reader of a terminal, and enters his ID*i* and PW*i*;
(2) The smart card firstly check the validity of the password by computing C*i*′= h(R*i*□h(PW*i*)) □h(PW*i*) and checking whether C*i*′=C*i*. If PW*i* is valid, user is permitted to enter the new password PW*i**. Otherwise, smart card rejects user's password change request.
(3) When receiving the new password, the smart card does the following computations：

R*i** = R*i*□h(PW*i*)□h(PW*i**) = h(ID*i*□x)□h(PW*i**)

C*i** = C*i*□h(PW*i*)□h(PW*i**) = h(h(ID*i*□x))□h(PW*i**)

(4) The smart card replaces R*i* and C*i* with R*i** and C*i** respectively.

4. Formal Security Proof Based on Strand Spaces

Strand spaces theory is a formal analysis method for security protocol, being presented by Fabrega, Herzog and Guttman. [26, 27] A Strand space is a collection of strands, equipped with a graph structure generated by causal interaction. In this framework, protocol correctness claims may be expressed in terms of the connections between strands of different kinds. Strand spaces model strictly standardizes authorized entity's behaviors, attacker's ability and operating environment. It can correctly describe the sequence and consequence of actions during the protocol process, and provide an effective analysis theory for protocol formal analysis.

In this section, we present the formal proof for the protocol based on the strand spaces theory. The correlative definitions and propositions of the basic strand spaces model can be referred to the references [26, 27].

4.1 The Extension of Strand Spaces

The trusted authentication protocol is designed based on Diffie-Hellman key exchange, and is not able to be described and analyzed by the basic strand spaces model. Therefore, we will extend the model for our protocol's formal analysis.

Definition 4.1 The set of term A consists of the following sets:

(1) $T \subset A$, T consists of predictable information;

(2) $N \subset A$, N is the ID of the protocol participant;

(3) $R \subset A$, R consists of unpredictable random numbers. R_p is the random number generated by the penetrator.

(4) $K \subset A$, K is a set of secret key. K_p is the set of keys which are held by the penetrator.

(5) $D \subset A$, D includes Diffie-Hellman values. D_p is the DH values held by the penetrator.

The intersection of each two sets (T, N, K, D) is empty set.

Definition 4.2 The operations of terms are the following three:

(1) Hash: A→A, describing hash function;

(2) $\|$: A×A→A, describing concatenation operation;

(3) $\oplus$: A×A→A, describing exclusive-or operation.

Definition 4.3 A penetrator trace is one of the following:

(1) **M.** Text message: $<+t>$ where $t \in T$;

(2) **C.** Concatenation: $<-g,\ -h\ , +gh>$;

(3) **S.** Separation into components: $<-gh,\ +g,\ +h >$;

(4) **K.** Key: $<+K>$ where $K \in K_p$;

(5) **F.** Intercepting DH: $<-d >$ where $d \in D_p$;

(6) **T.** Transferring after intercepting : $<-g,\ +g\ , +g>$;

(7) **H.** Hash function: $<-g,\ +hash(g)>$;

(8) **OX.** Exclusive-or operation: $<-g,\ -h\ , +g \oplus h>$;

(9) **R.** Generating random number: $<-r >$ where $r \in R_p$.

Definition 4.4 The trusted authentication protocol presented by us is defined as:

(1) Correspondence property: The both sides in the authentication protocol computes sharing secrets h(ID$\oplus$x), and uses it to do exclusive-or operation with DH value. Meanwhile, they use hash function h(.) to operate message authentication, and then generating a same session key.

(2) Secrete property: The both sides can obtain g^{NaNb}, which can't be computed by the penetrator.

4.2 Formal Proof for the Trusted Authentication Protocol

Definition 4.5 Supposing Σ is a strand space, then:

(1) Init[ID, x, g^{Na}, g^{Nb}] is an initiator strand with trace:

<+ID g^{Na} ⊕h(ID ⊕ x)h(g^{Na})， $-g^{Nb}$ ⊕h(ID ⊕ x) h(g^{Na} ‖ g^{Nb})， +h(g^{Nb})>, where ID∈ N, g^{Na} , g^{Nb} ∈ T, and g^{Na} ∉ N, x∈K. Σ_{init} is the set of initiator strands;

(2) Resp[ID, x, g^{Na} , g^{Nb}] is a responder strand with trace:

<−ID g^{Na} ⊕ h(ID ⊕ x)h(g^{Na})， $-g^{Nb}$ ⊕h(ID ⊕ x) h(g^{Na} ‖ g^{Nb})， +h(g^{Nb})>, where ID∈N, g^{Na} , g^{Nb} ∈T, and g^{Na} ∉ N, x∈K. Σ_{resp} is the set of responder strands.

Then, the strand space of the presented protocol is denoted as (Σ, P) = Σ_{init} ∪ Σ_{resp} ∪ P, where P is the set of penetrator strands. The strand space model of the protocol is illustrated as the figure 4.

$$\begin{tikzcd}
\text{Init[ID, x, } g^{Na}\text{, } g^{Nb}\text{]} & \text{Resp[ID, x, } g^{Na}\text{, } g^{Nb}\text{]} \\
\langle +IDg^{Na} \oplus h(ID \oplus x)h(g^{Na})\rangle \arrow[r] \arrow[d, Rightarrow] & \langle -IDg^{Na} \oplus h(ID \oplus x)h(g^{Na})\rangle \arrow[d, Rightarrow] \\
\langle -g^{Nb} \oplus h(ID \oplus x)h(g^{Na} \| g^{Nb})\rangle \arrow[d, Rightarrow] & \langle +g^{Nb} \oplus h(ID \oplus x)h(g^{Na} \| g^{Nb})\rangle \arrow[l] \arrow[d, Rightarrow] \\
\langle +h(g^{Nb})\rangle \arrow[r] & \langle -h(g^{Nb})\rangle
\end{tikzcd}$$

Figure 4. Strand space model of the presented protocol

1. Correspondence: the responder's and initiator's guarantee

Proposition 4.1 Suppose:

a) Σ is a RATP strand space, C is a bundle in (Σ,P), and r is a responder strand in Resp[ID, x, g^{Na} , g^{Nb}] with *C-height* 3.

b) ID∈N , x∉ R_p, $g^{Na} \neq g^{Nb}$, g^{Nb} is uniquely originating in Σ.

Then C contains an initiator strand s∈Resp[ID, x, g^{Na} , g^{Nb}] with *C-height* 3.

(1) **Lemma 4.1** g^{Nb} originates at the node <r, 2>.

Proof. Known as proposition 4.1, g^{Nb} ⊂uns_term(<r, 2>), and sign(<r, 2>)=+, the node <r, 1> is preceding <r, 2> on the same strand. So, we only need prove g^{Nb} ⊄uns_term(<r, 2>).

For uns_term(<r, 2>)= ID g^{Na} ⊕h(ID ⊕ x)h(g^{Na}), known as proposition 4.1 and the correlative definitions, $g^{Nb} \notin N \Rightarrow g^{Nb}$ ≠ID; $g^{Nb} \notin K \Rightarrow g^{Nb}$ ≠x; $g^{Na} \neq g^{Nb}$. So, g^{Nb} ⊄ g^{Na} ⊕h(ID ⊕ x), g^{Nb} ⊄h(g^{Na}).

Therefore, g^{Nb} ⊄ uns_term(<r, 1>), that is, g^{Nb} originates at the node *<r, 2>*.

(2) To the responder strand r = Resp[ID, x, g^{Na} , g^{Nb}],

Because g^{Nb} ⊂ term(<r, 2>), g^{Nb} ⊂h(g^{Nb}), and h(g^{Nb}) is a new element in the node *<r, 3>*, *<r, 2>*$\Rightarrow^{+}$*<r, 3>* is a converted edge of g^{Nb} . For the lemma 4.1, g^{Nb} is uniquely originates at the node *<r, 2>*.

According to the correlative definition, *<r, 2>*$\Rightarrow^{+}$*<r, 3>* is a test to g^{Nb} .

(3) **Lemma 4.2** Supposing n_0, n_1∈ Σ, and $n_0 \Rightarrow^{+} n_1$ is a test to g^{Nb} , If ∃t ⊂term(n_1), and the new element h(g^{Nb})⊂t, then there must be regular nodes m_0, m_1∈ Σ,where t ⊂term(m_1), and $m_0 \Rightarrow^{+} m_1$ is a converting edge of g^{Nb} .

Proof. Supposing there is not such a regular node m_1 that $m_1 \in \Sigma$ and $t \subset$ term(m_1), then, there must be a penetrator node $p \in \Sigma$ with positive sign, to which, t' is a new element and h(g^{Nb}) $\subset t'$. Considering all the possible traces of penetrator strands:

M. The trace has the form <+t> where $t \in T$. It is in contradiction to $t' \notin T$;

C. The trace has the form <$-g$, $-h$, $+gh$>, and thus $t' \subset g$ or $t' \subset h$. It is in contradiction to the hypothesis that t is a new element of p.

S. The trace has the form <$-gh$, $+g$, $+h$ >, and thus $t' \subset g$ or $t' \subset h$. It is in contradiction to the hypothesis that t' is a new element of p.

K. The trace has the form <+K> where $K \in K_p$. It is in contradiction to $t' \notin K_p$.

F. The trace has the form <$-d$> where $d \in D_p$. It is in contradiction to $t' \notin D_p$.

T. The trace has the form <$-g$, $+g$, $+g$>, and thus $t' \subset g$. It is in contradiction to the hypothesis that t' is a new element of p.

H. The trace has the form <$-g$, $+hash(g)$>, and thus $t' \subset g$. It is in contradiction to the hypothesis that t' is a new element of p.

OX. The trace has the form <$-g$, $-h$, $+g \oplus h$ >, and thus $t' \subset g$ or $t' \subset h$. It is in contradiction to the hypothesis that t' is a new element of p.

R. The trace has the form <+r> where $r \in R_p$. It is in contradiction to $t' \notin R_{p.}$

Therefore, the hypothesis is impossible. Then, there are regular nodes $m_0, m_1 \in \Sigma$, where $t \subset$ term(m_1) and h(g^{Nb}) $\subset t$. What's more, because there is not a node $m_1' \in \Sigma$ that preceding m_1, where $t \subset$ term(m_1') and h(g^{Nb}) $\subset t$. So, t is a new element of m_1.

In conclusion, $m_0 \Rightarrow^+ m_1$ is a converting edge of g^{Nb} .

(4) Concluded from (2), in the responder strand r = Resp[ID, x, g^{Na} , g^{Nb}], <r, 2>$\Rightarrow^+$<r, 3> is a test to g^{Nb} . According to the lemma 4.2, there must be regular nodes m_0, $m_1 \in \Sigma$,where $m_0 \Rightarrow^+ m_1$ is a converting edge of g^{Nb} . So, m_0 is a regular node with the sign of negative, and $g^{Nb} \subset$ term(m_0). Then, m_0 is in an initiator strand S', and m_0=< S', 2>, that is, term(< S', 2>)=$-$ g^{Nb} $\oplus$h(ID$\oplus$x)h(g^{Na} $\|$ g^{Nb}).

(5) Comparing term(< S', 2>) with the nodes in initiator strand, we can see there is only < S, 2> having the same form with < S', 2> in the initiator strand. So, ID′= ID, $g^{Na} = g^{Nb}$, x=x', and *C-height* of S is 3.

By now, we have proved the responder's correspondence property. And, in a similar way, the initiator's can be proved, too. That is, the correspondence property of the protocol is proved.

2. Secrecy

For being based on Diffie-Hellman key agreement, the protocol's secrecy is guaranteed by the secrecy of DH. g^{Na} , g^{Nb} are the sharing keys between the entities, which should be proved secure in the protocol. According to the computational difficulty of discrete logarithms, if g^{Na} , g^{Nb} originate from penetrator nodes, N_a, N_b must originate from them, too. So, if g^{Na} , g^{Nb} is secure, all the keys computed in responder or initiator will be secure.

Proposition 4.2 Suppose:

a) C is a bundle in (Σ,P), g^{Na} , g^{Nb} are uniquely originating in regular strand of C.

b) $g^{Na} \neq g^{Nb}$, $x \notin K_p$, and discrete logarithms is computationally infeasible.

Then N_a, N_b are secure absolutely.

Proof. Because g^{Na} is uniquely originating in regular strand of C, there must be a $s \in \Sigma_{init}$, where g^{Na} is uniquely originating in the node < S, 1>. And the form of < S, 1> is ID g^{Na} $\oplus$h(ID$\oplus$ x)h(g^{Na}).

Similarly, g^{Nb} is uniquely originating in regular strand of *C*, there must be a $s \in \Sigma_{resp}$, where g^{Nb} is uniquely originating in the node <*r, 2*>. And the form of < *r, 2*> is $g^{Nb} \oplus$h(ID$\oplus x$)h(g^{Na} ‖ g^{Nb}).

(1) Firstly, if the penetrator get $g^{Na} \oplus$h(ID$\oplus$x), he must get h(ID$\oplus$x) to compute g^{Na}. However, as the hypotheses, *x* is not in K_p. That is, it's impossible for penetrator to get h(ID$\oplus$x). Secondly, if the penetrator get h(g^{Na}), he can't obtain g^{Na} from it because of the irreversibility of Hash function. Even if he can get g^{Na}, he still can't get N_a because the computational difficulty of DH problem.

(2) Similarly, the penetrator can't get g^{Nb} or N_b, even if he gets $g^{Nb} \oplus$h(ID$\oplus$x) or h(g^{Na} ‖ g^{Nb}).

So, if only g^{Na}, g^{Nb} are originating in regular strand, N_a and N_b will keep confidentiality.

Proposition 4.3 Supposing *C* is a bundle in (Σ,*P*) and $x \notin K_p$, then the terms which contains g^{Na} and g^{Nb} must originate in regular strand. That is, these terms are impossible to originate in penetrator node.

Proof. In the protocol, there are two trace forms of the terms which contain g^{Na}, g^{Nb}: M$\oplus$h(ID$\oplus$ *x*) and h(M). Hereinto, M is representing g^{Na} or g^{Nb}.

(1) Supposing the term with the form as M$\oplus$h(ID$\oplus x$) is originating in a penetrator node, then the only possible trace of this penetrator strand *p* is $<-g,\ -h,\ +g \oplus h>$, and M$\oplus$h(ID$\oplus x$) originates in < *p, 3*>. Thus, h(ID$\oplus x$)$\subset$g, or h(ID$\oplus x$)$\subset$h. Further, because *x* must be obtained to compute h(ID$\oplus x$), M$\oplus$h(ID$\oplus x$) originating in penetrator node is in contradiction to $x \notin K_p$.

(2) Supposing the term with the form as h(M) is originating in a penetrator node, then the only possible trace of this penetrator strand *p* is $<-g,\ +hash(g)>$, and h(M) originates in < *p, 2*>. Thus, h(M)= *g* or h(M)$\subset g$. However, h(M)= *g* or h(M)$\subset g$ means h(M) originates in < *p, 1*>, which is in contradiction to the above conclusion that it originates in < *p, 2*>.

To sum up, all the terms containing g^{Na} or g^{Nb} must be originating in a regular strand.

According to the proposition 4.2 and 4.3, the secrets g^{Na} and g^{Nb} to be used to compute the sharing session key can keep confidentiality. That is, nobody but the both sides of the protocol can compute g^{NaNb}.

By now, the secrecy of the protocol is proved.

5. Informal Security Analysis

In fact, the security of the proposed trusted authentication protocol is guaranteed by its mutual authentication and key agreement phase. If this phase is completed securely, the sharing session key can protect the following integrity attestation. So, in this section, we firstly examine the security of the proposed mutual authentication and key agreement scheme using SDIO smart card from the following aspect.

(1) **The proposed protocol resists the privileged insider's attack**. In the registration phase, h(PW*i*) is submitted, instead of submitting password in plain text form. Thus, the privileged insider of the server can't obtain the password. Therefore, the scheme can withstand the privileged insider's attack.

(2) **The proposed protocol resists the replay attack.** The authentication is based on challenge and response, which decides that a replay attack can't pass the subsequent challenges.

(3) **The proposed protocol resists parallel session attack.** The parallel session attack is impossible to occur because the challenge values of the both sides never appear in plain text form during the whole authentication process. An attacker can't acquire any valid message to masquerade a legal user or a remote server.

(4) **The proposed protocol resists guessing attack.** It resists online guessing attack because entering wrong password is limited to three in the system. For the offline guessing attacks, even if the

attacker get R*i* = h(ID*i*□x)□h(PW*i*) stored in the smart card, he can't obtain the ID*i* or PW*i* because of the protection of Hash.

(5) **The proposed protocol provides fast wrong password detection.** If user U*i* inputs the wrong password by mistake, this wrong password will be quickly detected by the smart card since the smart card can verify C*i*'= h(R*i*□h(PW*i*))□h(PW*i*) using the stored C*i* in the login phase and stop the following information exchange in time.

(6) **The proposed protocol provides secure password change.** Because the smart card verifies the old password firstly in the password change phase, when a smart card is stolen, unauthorized users can't change the password of the card. Thus, no one can perform the Denial of Service attack using a stolen card.

(7) **The proposed protocol provides perfect forward secrecy.** The key agreement of the proposed scheme uses Diffie-Hellman key agreement scheme, which is well known to be able to provide perfect forward secrecy.

(8) **The proposed protocol has no time synchronization problem.** The scheme uses random numbers, not time stamp to be the challenge values. So, there is no time synchronization problem.

(9) **The proposed protocol has integrity attestation.** In the scheme, not only the identity but also the integrity of the user is authenticated by the server. This guarantees the terminal security in a DRM system.

(10) **The proposed protocol's correctness has been formally proved.** As demonstrated in the section 4, the protocol was proved to be correct using strand spaces theory.

Here, we compare its security properties with the other related protocols [20, 22, 23, 24, 25]. Table 1 shows the comparison results.

Table 1 Security properties of the proposed protocol with other related protocols

Security properties	**Shieh-Wang's [20]**	**Wang et al. [22]**	**Yang et al. [23]**	**Xu et al. [24]**	**Yoon [25]**	**Proposed protocol**
privileged insider's attack	Insecure	Secure	Secure	Insecure	Secure	Secure
replay attack	Secure	Secure	Secure	Secure	Secure	Secure
parallel session attack	Secure	Secure	Secure	Secure	Secure	Secure
guessing attack	Secure	Insecure	Secure	Secure	Secure	Secure
fast wrong password detection	No	Yes	No	No	Yes	Yes
secure password change	No	Yes	No	No	Yes	Yes
perfect forward secrecy	No	No	Yes	No	Yes	Yes
no time synchronization problem	Yes	No	Yes	No	Yes	Yes
no verification table	Yes	Yes	Yes	Yes	Yes	Yes
mutual authentication	Yes	Yes	Yes	Yes	Yes	Yes
integrity attestation	No	No	No	No	No	Yes
formal security proof	No	No	No	Yes	No	Yes

6. Performance Comparisons

This section analyzes the efficiency of the proposed protocol. Table 2 provides computational costs of the proposed protocol with the above five protocols in regards to the registration, login, authentication and key agreement. Here we consider notation T_h, T_A, T_{MA} and T_{ME} as the computational cost of one way hash function, asymmetric encryption, modular addition, and modular exponentiation, respectively.

Table 2 Computational costs of the proposed protocol with other related protocols

	Registration phase	**Login phase**	**Authentication and key agreement phase**	**Password change phase**
Shieh-Wang's [20]	$1T_h$	$1T_h$	$8T_h$	No support
Wang et al. [22]	$3T_h$	$4T_h$	$4T_h$	$4T_h$
Yang et al. [23]	$5T_h$	$1T_h + 1T_{ME}$	$3T_{ME}+4T_A$	$2T_h$
Xu et al. [24]	$2T_h+1T_{ME}$	$3T_h+2T_{ME}$	$6T_h+4T_{ME}$	No support
Yoon [25]	$3T_h$	$3T_h$	$6T_h+4T_{MA}$	$4T_h$
Proposed protocol	$3T_h$	$2T_h + 1T_{ME}$	$6T_h+5T_{ME}$	$5T_h$

As in table 1 and table 2, we can see that the proposed protocol not only provides more security assurances, but also has the reasonable computational costs. What's more, the proposed protocol is formally proved to be secure, and is the only one that provides integrity attestation.

7. Conclusion and Future Prospects

This paper demonstrated a trusted authentication protocol for DRM system based on mutual authentication and key agreement scheme using SDIO smart card. Together with the protocol, a strict formal proof was presented, too. What's more, in contrast to other corrective schemes through attack resisting analysis and computational cost analysis, the proposed scheme is able to provide greater security and practicality.

Acknowledgments The work was sponsored by the National Natural Science Foundation of China (Grant No.61003234), Program for Science & Technology Innovation Talents in Universities of Henan Province (Grant No.2011HASTIT015), and Plan For Scientific Innovation Talent of Henan Province （Grant No.134100510011）.

8. References

[1] Zhang Zhiyong, “Digital Rights Management Ecosystem and its Usage Controls: A Survey”, International Journal of Digital Content Technology & Its Applications, vol.5, No.3, pp.255-272, 2011.

[2] Lili Zhang, Zhiyong Zhang, Danmei Niu, Tao Huang, "A Novel DRM Security Scheme and its Prototype System Implementation", International Journal of Digital Content Technology & Its Applications, Vol. 5, No. 11, pp. 334 ~ 342, 2011

[3] Stamm S, Nicholas Paul Sheppsrd, “Implementing trusted terminals with a TPM and SIDDRM”, In Proceedings of REM 2007, pp.73-85, 2007.

[4] Sadighi A R, Wolf M, “Christisn stuble enabling fairer rights management with trusted computing”, In Proceedings of ISC 2007,pp. 53-70, 2007.

[5] Gallery E. Authorisation Issues for Mobile Code in Mobile Systems, London: Royal Holloway, University of London, 2007.

[6] Trusted Computig Group. TCG Specification Architecture Overview. Specification Revision1.2. 2004, http://www.trustedcomputinggroup.org/specs/IWG/TCG_1_0_Architecture_Overview.pdf.

[7] Reiner Sailer, Xianlan Zhang, Trent Jaeger, and Leendert van Doorm, “Design and implemention of a TCG-based integrity measurement architecture”, In proceedings of the 13th USENIX Security Symposium, pp.223-238, 2004.

[8] Frederic Stumpf, Omid Tafreschi, Patrick Röder. “A Robust Integrity Reporting Protocol for Remote Attestation”, IBM research, 2006.

[9] Hailun Tan, Wen Hu, Sanjay Jha, “A TPM-enabled Remote Attestation Protocol(TRAP) in Wireless Sensor Networks”, In proceedings of the 6th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, pp.9-16,2011.

[10] Gao zhigang, Feng dengguo, “Efficient Identity-Based Authenticated Key Agreement Protocol in the Standard Model”, Journal of Software, vol.22, No.5, pp.1031-1040, 2011.

[11] Kenneth Goldman, Ronald Perez, Reiner Sailer, “Linking remote attestation to secure tunnel endpoints”, In proceedings of the first ACM workshop on Scalable trusted computing (STC’06), pp.21-24, 2006.

[12] Gasmi Y, Ahmad-Reza S, Patrick S, et al, “Beyond Secure Channels”, In proceedings of the 2nd ACM Workshop on Scalable Trusted Computing (STC 2007), pp.30-40, 2007.

[13] Zhou Lingli, Zhang Zhenfeng, “Trusted Channels with Password-based Authentication and TPM-based Attestation”, In proceedings of International Conference on Communication and Mobile Computing, pp.223-227, 2010.

[14] Ahmad-Reza Sadeghi, Steffen Schulz, “Extending IPsec for Efficient Remote Attestation”, Lecture Notes in Computer Science, vol.6054, pp.150-165, 2010.

[15] Jian Wang, Haihang Wang, Chengxiang Tan, “Cyptanalysis and Improvement of An ‘Efficient Remote Mutual Authentication and Key Agreement’”, In proceedings of 2008 IEEE Adia-Pacific Services Computing Conference, pp.835-840, 2008.

[16] Lamport, L., “Password authentication with insecure communication”, Communication of ACM, vol.24, No.11, pp.770-772,1981.

[17] H.Y. Chien, J.K.J., and Y.M. Tseng, “An efficient and practical solution to remote authentication: smart card”, Computers & Security, vol.21, No.4, pp.372-375, 2002.

[18] Hsu, C.L., Chien et al, “Remote user authentication scheme using smart cards”, Computer Standards and Interfaces, vol.26, No.3, pp.167-169, 2004.

[19] Juang, W.S., “Efficient password authenticated key agreement using smart cards”, Computers and Security, vol.23, No.2, pp.167-173, 2004.

[20] J.M.Wang, W.G.S.a., “Efficient remote mutual authentication and key agreement”, Computers and Security, vol.25, No.1, pp.72-77, 2006.

[21] Yoo, E.-J.Y.a.K.-Y, “Two security problems of efficient remote mutual authentication and key agreement”, In Proceedings of Future Generation Communication and Networking (FGCN 2007), pp. 66-70, 2007.

[22] X.M. Wang, W.F. Zhang, J.S. Zhang, M.K. Khan, “Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards”, Computer Standards & Interfaces, vol.29, No.5, pp.507-512, 2007.

[23] G. Yang, D. S. Wong, H. Wang, X. Deng, “Two-factor mutual authentication based on smart cards and passwords”, Journal of Computer and System Sciences, vol.74, No.7, pp.1160-1172, 2008.

[24] J. Xu, W.T. Zhu, D. G. Feng, “An improved smart card based password authentication scheme with provable security”, Computer Standards & Interface, vol.31, pp.723-728, 2009.

[25] Eun-Jun Yoon, “Remote mutual authentication and key agreement scheme based on elliptic curve cryptosystem”, Turk J Elec Eng & Comp Sci, vol.19, No.3, pp.335-347, 2011.

[26] Fabrega, F.J.T.H., J.C. Guttman, J.D., “Strand spaces: why is a security protocol correct?”, In Proceedings of 1998 IEEE Symposium on Security and Privacy, pp. 160-171,1998.

[27] Fábrega, F.J.T., “Strand spaces: proving security protocols correct”, Journal of Computer Security, vol.7, No.2-3, pp.191-230, 1999.