Mutual Authentication Scheme with Smart Cards and Password under Trusted Computing

Li Yang, Jian-Feng Ma, and Qi Jiang
(Corresponding author: Li Yang)

School of Computer Science and Technology, Xidian University,
No.2 South Taibai Road, Xi'an, Shaanxi, 710071, China
(Email: xdyangli@gmail.com)
(Received Oct. 12, 2010; revised and accepted Jan. 25, 2011)

Abstract

Only identities of the server and the user are authenticated in traditional smart cards based password authentication schemes, but the platform does not be verified, and which cannot provide enough protection on personal information of the user. A mutual authentication scheme based on smart cards and password is proposed under trusted computing, in which hash functions are used to authenticate identities, and remote attestation is used to verify the platform. Analysis showed that our scheme can resist most of the possible attacks, is secure and efficient, and fulfills the designed security goals, such as secure session key agreement, user identity anonymity, password free changing, platform certification updating.

Keywords: Authentication, password, remote attestation, smart card, trusted computing

1 Introduction

Smart card based password authentication method provides two-factor of user authentication mechanisms which is based on smart card and password each. Since Lamport [5] introduced the first well-known password based remote user authentication scheme using smart card, there have been many smart card based password authentication schemes proposed, some recent ones are [1, 6, 7, 8, 18, 19]. A typical smart card based password authentication scheme consists of three phases that is registration phase, login phase and authentication phase. In registration phase, a user sends registration request information to the server in secure channel, then the server issues and replays to the user registration information generated by user's identity and the long time secure key of the server himself; in login phase, the user inputs his smart card into a terminal, then enters the identity and password to getting login permissions, the terminal constructs login request message using password and the information in the smart card and sends it to the remote server; in authentication phase, the server checks the legitimacy of the login request using its long-time key and etc. to complete the authentication of the user, and the server sends its message back to the user, then the user checks the legitimacy of the message using password and registration information by the server to complete authentication of the server. After mutual authentication, the user sends his personal information to the server and accepts the server's service.

But the above typical scheme has some obvious deficiencies with increasingly openness and more complex network environment. Only the identities of the smart card user and the server are authenticated when request login in these schemes, and their platforms do not be checked. So when the user platform are not security enough, it may lead potential hazards on the server, while the server platform is not security enough, the user information can not be protected effectively, it may lead to the disclosure of user information, etc. There are many different kinds of malicious software in openness network environment, such as Trojan Horse, Worms and etc. Once the server fails to agree in a particular configuration of state security, for example its operating system not installed the latest patches, existed software security vulnerabilities, software version expired, controlled with malicious software, which will caused security hazards to users login the server, and it may result in disclosure to logged on users of personal information to request failed business deal, property losses. So, it required to design a new smart card based password authentication scheme to meet the new objectives when users login to the sever remotely in the Internet environment. For this problem, [4] provides a mutual authentication scheme between smart card users and a trusted terminal platform. Using this scheme, it can ensure the terminal platform of users is trusted, but the problem when a smart card user and his platform as a whole login to the server are still not be resolved.

In this paper, we propose a new mutual authentication scheme using smart card and password under trusted

computing, for the purpose to resolve the shortcoming that the trusted attribution of the server platform not been checked in traditional smart card based password authentication schemes. And in our scheme, we provide mutual authentication of identity, verifying of the server platform property, and secure session key agreement, providing much security protection for smart card users and the server, supporting user identity anonymity, supporting user password free changing and server platform trusted certificate updating. The analysis shows that our scheme is secure and efficient, and can resist common attacks to the scheme, such as Dos attack, password guessing attack, forgery attack, parallel attack and platform impersonation attacks and so on, meets the designed goals.

The remainder of the paper is organized as follows. In Section 2, we introduce trusted computing and remote attestation. In Section 3, give the security goals and security requirements. In Section 4, descript of a detailed implementation of the scheme. In section 5, analyze security property and performance of the scheme. Finally, in Section 6, we conclude the paper.

2 Background Knowledge

2.1 Trusted Computing and Remote Attestation

Trusted computing (TC) technology came up with Trusted Computing Group (TCG) [14], its technical specification is accepted by the academic and industry, and becoming a research hotspot recently [9, 10, 11, 12]. The core of TC is embedding a chip called Trusted Platform Module (TPM) [15] into the platforms. As a relative independent security co-processor, TPM can provide encryption function and protected storage; provide hardware assistance for mechanism and security function. TPM is the base for measuring and validating the trusted attribution for the platform also.

TPM offers the possibility to attest the configuration of the local platform to a remote verifier which is called Remote Attestation (RA). In remote attestation the integrity of platform configuration is verified by the remote verifier. There are a set of Platform Configuration Registers (PCRs) in TPM, in which the integrity measurement values are stored. When powered on, the PCRs are initialized, the hardware and software modules of the platform are measured, and the corresponding hash values are stored in PCRs, the measurement event are created, and the record is logged in the Stored Measurement Log (SML) while measured. Both the PCRs and SML are used to attest the platform's security state to the remote verifier. To ensure that the measurement value is true, TPM uses its Attestation Identity Key (AIK) to sign the measurement value.

In remote attestation, the computing platform responses the request of the remote verifier, collects the event log, PCRs of platform, and signature on them by using AIK private key, and then sends these message to the remote verifier; the remote verifier validates the submitted information, and then checks the computing platform identity and reported information, judges whether the platform is trusted. The more detailed process is the authenticating party sends attestation request to computing the platform, the replied messages of the computing platform includes the current measurement list and PCRs, then TPM in the computing platform signs it; the authenticating party recalculate PCRs using the measurement list, and verifies the consistent of the result set value with the signature of the reference value; the authenticating party compares the measurement list with the existing data, and verifies the computing platform identity and its integrity.

3 Security Requirements and Definitions

3.1 Security Requirements

In this section, we define the security goals a smart card based and password authentication scheme should achieve. In addition, we shall also introduce all of the attacks that a smart card based and password authentication scheme should withstand. Following with [2] and [17], we resort them under trusted computing as follows.

G1: The scheme should provide mutual identity authentication.

G2: The password cannot be revealed by the administrator of the server.

G3: A session key is established during the password authentication process to provide confidentiality of communication.

G4: The password can be chosen and changed freely by the user.

G5: The user can login the server anonymously.

G6: The platform of the server can be verified.

G7: The integrity certificate of the server can be updated correctly.

A smart card based and password authentication scheme should with stand all of the attacks bellow under trusted computing.

S1: Denial of Service Attacks

S2: Replay Attacks

S3: Password Guessing Attacks

S4: Parallel Session Attacks

S5: Forgery Attacks

Table 1: Symbol notations

Symbol	Notations
S	the server
U	the user
ID	the identity of a user
PW	the password of U
$h()$	a one-way hash function
$E_K()$	Encryption of a message by K
$\oplus$	XOR operation
T	time stamps
AIK_{priv}	public AIK of the platform
AIK_{pub}	public AIK of the platform
$Cert_{AIK}$	AIK certification of the platform
$Sig\{X\}_{AIK}$	AIK signature
$Log(X)$	security measurement log extraction

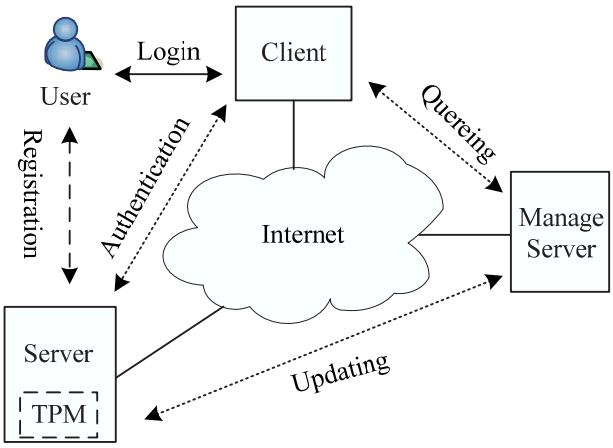


Figure 1: Prototype of the proposed scheme

S6: Inner Attacks

S7: Stolen-verifier Attacks

S8: Platform Impersonation Attacks

3.2 Definitions

The symbol notations of our scheme are as shown in Table 1.

4 Trusted Mutual Authentication Scheme under Trusted Computing

We have implemented our scheme along with the prototype depicted in Figure 1. In this prototype, the client, the server (with a TPM chip), and the management server connected each other via the wired Internet. The smart card user registers at the server firstly, and then chooses a right client to login and sends access request messages to the server. After received the messages, then the server will complete mutual authentication with the user. The management server provides service of updating to the server platform certification and permits the user to inquiring from.

The process above includes three phases: register phase, login and authentication phase, and update phase. From [4], we suppose that the platform of the user is trusted, and it needs not to verify the integrity of the client platform in our scheme.

4.1 Registration Phase

Firstly, the user needs to become a legal registered user of the server. The registration process includes two steps: the user registration requirement information sending to the server and the issued information sending back from the server to the user. Detailed steps are as follows.

1) When a user U sends his register requirement messages to the server S, U arbitrarily chooses a unique identity ID and password PW, then calculates $h(PW)$ and sends ID, $h(PW)$ to S.

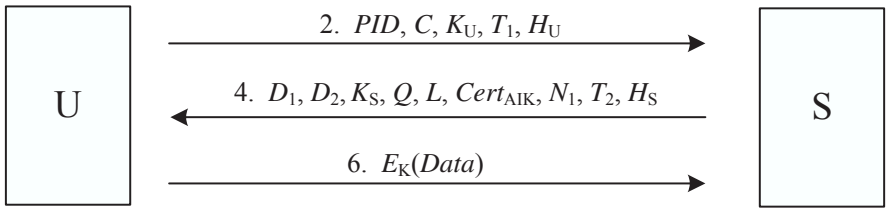
2) After received registration requirement messages from U, the server S calculates $PID = h(x, ID)$, $I = h(Cert_{AIK})$, $B = PID \oplus h(PW) \oplus I$, where x is a secret number which is selected by S randomly, and is greater than 100 bits for security considerations. $Cert_{AIK}$ is the platform AIK certification of S. And S generates a large prime number p and $g \in GF(p)$, chooses a random number N_0, and then issues PID, B, I, N_0, p, g to U as his registration information by a secure channel.

4.2 Login and Authentication Phase

When a user U sends a login request to the server S using by his smart card issued by the server on a remote client, S performs the following steps as depicted in Figure 2.

1) The user U inserts his smart card into a client and enters ID and PW the smart card and the client calculate $h(PW)$ and check whether ID and PW are valid. While ID and PW are all right, the client calculates $C = h(B \oplus h(PW) \oplus N_0 \oplus T_1)$, where T_1 is the time-stamp. U generates a secret number a, calculates $K_U = g^a \bmod p$, $H_U = h(PID, C, KU, T_1)$, and sends message PID, C, KU, T_1, HU to S.

2) Upon receiving the message from U, S checks the time stamp T_1, checks if $T_1' - T1 \leq \Delta T$, where T_1' is the current time stamp of S, ΔT is the legal time interval for transmission delay. If it is right, S calculates $H_U' = h(PID, C, K_U, T + 1)$, and checks if $H_U' = H_U$ for judging the message integrity by U. If



1. Calculation:
   $C=h(B \oplus h(PW) \oplus N_0 \oplus T_1)$;
   $K_U=g^a \bmod p$;
   $H_U=h(PID, C, K_U, T_1)$.
5. Verify: H_S; D_2; I; PCR.
   Calculation:
   $N_1= D_1 \oplus h(PID \oplus N_0 \oplus I \oplus T_2)$;
   $K_{US}=(K_S)^a=(g^b)^a \bmod p$.
3. Verify: T_1; H_U; C.
   Calculation:
   $D_1=h(PID \oplus N_0 \oplus I \oplus T_2) \oplus N_1$;
   $D_2=h(D_1 \oplus T_2)$; $K_S=g^b \bmod p$;
   $K_{US}=(K_U)^b=(g^a)^b \bmod p$;
   $PCR=\mathrm{SHA1}(PCR_0||\ldots||PCR_N)$;
   $Q=Sig\{PCR, h(K_S, N_1)\}_{AIK}$;
   $L=Log(SML)$;
   $H_S=h(D_1, D_2, K_S, Q, L, Cert_{AIK}, N_1, T_2)$.

Figure 2: Login and authentication phase

they are equal, then S verifies the identity of U, it is to verify whether PID and C are all correct. S calculates $C' = h(PID \oplus N_0 \oplus I \oplus T_1)$ and checks if C' equals to C, while they are equal, S knows U as the legitimate user of the system.

S calculates $D_1 = h(PID \oplus N_0 \oplus I \oplus T_2) \oplus N_1$, $D_2 = h(D_1 \oplus T_2)$, where T_2 is a time stamp generated by S. And S generates a secret number b, calculates $K_S = g^b \bmod p$, $K_{US} = (K_U)^b = (g^a)^b \bmod p$ by the secret key received from U. Then S loads AIK private key AIK_{priv} from its TPM, calculates integrity value PCRs of the server platform as $PCR_S = SHA1(PCR_0|| \ldots ||PCR_N)$, signs it with AIK_{priv} as $Q = Sig\{PCR_S, h(K_S, N_1)\}_{AIK}$, where N_1 is a random number chose by S, loads security measurement log as $L = \log(SML)$. Then S calculates $HS = h(D_1, D_2, KS, Q, L, Cert_{AIK}, N_1, T_2)$ and sends message D_1, D_2, KS, Q, L, $Cert_{AIK}$, N_1, T_2, HS to U.

3) When receiving message, U firstly calculates $T_2' - T2 \leq \Delta T$ and checks the correctness of time stamp T_2, where T_2' is the current time stamp of the user client. If it is right, U calculates $H_S' = h(D_1, D_2, KS, Q, L, Cert_{AIK}, N_1, T_2)$, and checks if H_S' equals to HS. If not, U rejects. Otherwise, U calculates a secret number $N_1 = D_1 \oplus h(PID \oplus N_0 \oplus I \oplus T_2)$ by using message who received, and stores N_1 for next login. Then U calculates $D_2' = h(D_1 \oplus T_2)$, $I' = h(Cert_{AIK})$ and checks if $D_2' = D_2$, $I' = I$. If they are all equal, U confirms the identity of S.

   After that, U needs to verify the integrity of the server platform, to ensure whether the configuration of S platform meeting the safety strategy and can be trusted. Firstly, U decrypts Q using AIK public key in the AIK certification $Cert_{AIK}$, gets PCR_S and $h(K_S, N_1)$, can verify K_S and N_1 again. Using message in L, U calculates $PCR_S' = SHA_1(PCR_0|| \ldots ||PCR_N)$, checks whether $PCR_S' = PCR_S$. If they are equal, the server platform is verified. Then, U calculates the session key as $K_{US} = (K_S)^a = (g^b)^a \bmod p$ using K_S which S sends to, and stores K_{US}.

4) By now, S verifies identity of U, and U authenticates identity and platform integrity of S. Then S considers U as a legal user and provides services for him, and the message translating between them is protected by the session key K_{US}.

4.3 Update Phase

The update phase of our scheme includes the user password updating and the server platform certification updating.

1) User password updating: In our scheme, U can change his password freely, while need not to interact with S. U inserts his smart card and key in a new password PW', calculates $B_1 = PID \oplus h(PW') \oplus I$. Then the smart card replaces B with B_1. When U login next time, he can use PW' and N_1 to generate new accessing requirement message.

2) Server platform certification updating: Under trusted computing, TPM will become a target for attackers, while be compromised or the identity leaked, the trusted property of platform can not be guaranteed. For updating his AIK certification, S replaces $Cert_{AIK}$ with $Cert'_{AIK}$, calculates $I = h(Cert'_{AIK})$ and submits it to the management server and notifies to users at the same time, then finishes the updating to I. Then S will publish the compromised

TPM through a trusted third party the management server. Before registration or login, U may query the new AIK certification of S from the management server.

5 Analysis

In this section, security analysis of the proposed scheme is given. We will show that the proposed scheme can withstand the various possible attacks and fulfills the designed security goals.

5.1 Security Goals Achieving

Our scheme provides mutual authentication between U and S, allows U changing password freely no case S is online or not, and allows S updating his AIK certification. Obviously, our scheme meets the security goals G1, G4 and G7. In the scheme, the user U uses anonymous identity PID to login, during the authentication S and U use known parameters p, g and use the DH key exchange protocol [3] to consult session key K_{US}, which fulfills forward security. The scheme meets the security goals G3 and G5. Other security goals are analysis as below.

1) In registration phase, U submits ID and $h(PW)$ to S, where PW is chose by the user, the plain text of PW is not included in message and $h()$ is a strong one way function. In login and authentication phase, the user does not send PW plaint text or $h(PW)$ directly instead of by calculating of C. In password changing phase, the user chooses his new password PW' by his own, and replaces B with $B1$ without the need for the participation of the server. In the whole scheme, the server manager can not obtain plant text of PW and the user password may not leak to the server manager, the scheme meets the security goal G2.

2) In login and authentication phase, while U authenticates identity of the server S, and authenticates the sever platform information Q, L, and $Cert_{AIK}$ at the same time, where PCR_s in L and Q is calculated from trusted measurement root, it is on behalf of the integrity of the server platform, Q is signed by TPM with AIK private key AIK_{priv}, they represent the server platform identity together with $Cert_{AIK}$. The verifying of them proves the server platform is trusted, the scheme meets security goal G6.

5.2 Security Analysis

1) In our scheme, the user needs to input the right ID and PW and go through legitimacy validation by the smart card, then can he/she make accessing requests to the server. At first, the attacker can not pass through the verification of the smart card, so he can not launch denial of service attacks to the server. Our scheme can resist this type of attacks and fulfills S1. Obviously, in login and authentication phase, the message that the user sends to the server includes a time stamp T_1, and the using of T_1 can help to resist message reply attacks. Our scheme fulfills S2.

2) Guessing attacks can be classified as on-line or offline guessing attacks. In our scheme, when using smart card, the user needs to submit correct ID and PW the attacker forgeries a legitimate user only if guessed the user ID and PW successfully. The on-line guessing attacks to our scheme can be prevented easily by limiting the number of failed login, so it will not be discussed here. In our scheme, the message which the user translating in network does not include plain text of PW the attacker can intercept only partial of exchanging data between the server and the user, and stores it for offline guessing attacks. In registration phase, the issued message which the server sends to the user is translated in a secure channel, which can prevent the attacker intercept the message illegally. Even if the attacker intercepted the message PID, C, and T_1 in Step 2 of Figure 2, where $PID = h(x, ID)$ and C is calculated by the user as $C = h(B \oplus h(PW) \oplus N_0 \oplus T_1)$ or $C = h(PID \oplus I \oplus N_0 \oplus T_1)$. Because the secret number x is included in PID, there is no plaintext password information in the message, and N_0 is a secret number issued by the server through a secure channel. So the attacker will not guess the user password by the message intercepted, our scheme can resist offline guessing attacks to the user password and fulfills S3.

3) If the attacker attempts parallel session attacking and wants to masqueraded as a legitimate user, he needs to use login message leaked by user when login server for not know about the password PW. In our scheme, the attacker cannot get anything about the user password PW from interactive session message between S and U. And only can the attacker use message eavesdrop from login and authentication phase by the user and to forge legitimate login and authentication message. In Step 2 of Figure 2, the message that the user sends to the server include KU and T_1. In Step 4 of Figure 2, the message that the server relays to the user include K_S, Q, T_2 and N_1. K_U and K_S are used by the server and the user to negotiate session key, Q is a signature value by the server's private AIK key signing on PCR and $h(K_S, N_1)$, this signature cannot be forged because of AIK. So the attacker cannot use the intercepted messages from one side to replay to the other, the scheme can prevent parallel session attacks. Besides, the use of timestamps strengthened to prevent such attacks. So, our scheme can resist parallel session attacks and fulfills S4.

4) If the attacker impersonates as a legal user and attempts to login the server, and when accesses the client terminals at the login stage, the attacker can-

not provide correct ID and PW the forgery attacks failed. Suppose that the attacker intercepted message from Step 2 in Figure 2 and relayed it for masquerading as a legitimate user, the attacker needs to guess and calculate C as $C_g = h(PID \oplus N_0' \oplus I \oplus T_1)$, where I is known by the attacker in advance step of the scheme, N_0' is guessed by the attacker. Clearly, the guessing value C_g by the attacker cannot go through the verification of C from Step 3 in Figure 2, the attack will fail. Suppose that the attacker has a strong computing power and guesses N0 successfully by offline dictionary guessing attacks. Now, the attacker can provide correct C, but the attacker cannot know about secret number a, in rest steps the attacker cannot finish session key agreement of K_{US} with S, the attack will failure. And suppose that the attacker intercepted message from Step 4 in Figure 2 and relayed it to masqueraded as a legitimate server, because the attacker know nothing about the secret number b of the server, in rest steps the attacker cannot finish session key agreement with U, the attack will fail. Our scheme can withstand impersonation attack and fulfills S5.

5) In our scheme, the user password list and verification list need not store in the server, the server knows secret number x, the others cannot change the user password PW besides the user, and this feature can prevent verifier stolen attacks and modify attacks. In registration phase, the server manager may guess the user password PW through offline guess attacks on $h(PW)$, our scheme cannot resist this type of attacks. But in most cases, the server will encrypt the user registration information, the other inner staff cannot get the user registration information besides the server manager. In the latter stages of communication, PW is not used directly. So the other inner staff will not get the user password PW and can not impersonate legitimate users to login the server. Our scheme can prevent inner attacks and stolen-verifier attacks, fulfills S6 and S7.

6) There is a type of attack called platform impersonation [13] under trusted computing. Suppose that the attacker controlled two servers, one is a trusted sever, the other is a malicious server. The attacker may launch these two servers together, and bypasses the remote attestation of the trusted server, by using the platform configuration of the honest client to attest the malicious server. In our scheme, during the remote attestation of the server to the user client, the using of the time stamp T_2 makes the attacker cannot replay message by the malicious server. At the same time, during session key agreement between S and U, S uses AIK private key AIK_{priv} signing on the hash value of K_S and N_1, while the attacker not know about AIK_{priv}, so he cannot do complete the signature. Our scheme can prevent the attacker to faking on the server platform information, can resist platform impersonation attacks, and fulfill S8.

Table 2: Security goals analysis

	G1	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**
Chien [1]	Y	Y	N	Y	N	N	/
Ku [6]	Y	Y	N	Y	Y	N	/
Yoon [19]	Y	Y	Y	Y	N	N	/
Liao [8]	Y	Y	Y	Y	N	N	/
Yang [18]	Y	Y	Y	Y	Y	N	/
Kumar [7]	Y	Y	Y	Y	Y	N	/
Ours	Y	Y	Y	Y	Y	Y	/

Table 3: Analysis of common attacks against

	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
Chien [1]	Y	Y	Y	N	Y	N	Y	N
Ku [6]	Y	Y	Y	N	Y	Y	Y	N
Yoon [19]	N	Y	Y	N	Y	Y	Y	N
Liao [8]	Y	Y	Y	N	Y	Y	Y	N
Yang [18]	Y	Y	Y	N	Y	Y	Y	N
Kumar [7]	Y	Y	Y	Y	Y	Y	Y	N
Ours	Y	Y	Y	Y	Y	Y	Y	Y

5.3 Performance Analysis

1) **Security performance:** We compare our scheme with other smart card based and password authentication schemes in security goals and resisting of common attacks. The results are showed in Table 2 and Table 3. Where Y means Supported, N means Not supported, "/" means Not involved.

   Besides achieved general security goals of smart card based password authentication scheme, the verification of the server platform is provided in our scheme. It enhanced performance and security of the scheme's resistance against attacks, and can give better protection to the smart card user from malicious servers or those servers do not meet the security policy of the potential hazards. From Table 2, we can see that compared with other schemes, our scheme can achieve more security goals. And from Table 3, we can see that compared with other schemes, our scheme also has a distinct advantage against common security attacks.

2) **Computation performance:** The calculation of the time complexity appropriate to meet the trusted computing environment certification requirements. The computation symbols of our scheme is as such below, t_h means one time of hash computation, t_{xor} means one time of XOR, t_{ek} means one time of encryption or decryption, t_{exp} means one time of modular exponentiation, t_{sig} means one time of signature,

t_{pk} means of one time of public key encryption and decryption operation, t_{PCR} means calculation of the platform PCR_s, t_{log} means one time of the platform security measurement log loading.

With analysis, the total time complexity of our scheme is as $17t_h + 19t_{xor} + 4t_{exp} + 1t_{sig} + 1t_{pk} + 2t_{PCR} + 1t_{log}$. In details, the registration phase is 3th+2txor, the login and authentication phase is $12t_h + 15t_{xor} + 4t_{exp} + 1t_{sig} + 1t_{pk} + 2t_{PCR} + 1t_{log}$, the updating phase is $2t_h + 2t_{xor}$. In our scheme, TPM in the server is as an independent computing unit and can complete partial of the calculation. By analysis, the computation by CPU is $17t_h + 19t_{xor} + 4t_{exp} + 1t_{pk} + 1t_{PCR}$, the computation by TPM is $1t_{sig} + 1t_{PCR} + 1t_{log}$, where t_{PCR} and t_{log} can be pre-computed by TPM, and its time complexity does not impact on the scheme.

Compared with other proposed schemes, when validating of the server platform, the time cost of the user terminal is increased by $1t_{pk} + 1t_h + 1t_{PCR}$, where the verification of PCR needs several times of hash operation, the increased time complexity does not affect the performance of the user platform or the overall performance of the scheme. The decryption of Q needs $1t_{pk}$, and Q is the AIK signature of PCR and $h(K_S, N_1)$. The length of PCR is $160bits$, $h(K_S, N_1)$ is not more than $256bits$, the time required for encryption and decryption is within the permissible level. And the increased time complexity provides the verification of the server platform, enhances the security of the scheme.

6 Conclusions

Using smart card and password for user authentication in openness network is a common security mechanism. Only the user identity is authenticated in the traditional smart card based password authentication schemes, but not the platform. In this paper, we proposed a new smart card based password authentication scheme under trusted computing environment. The proposed scheme can provide mutual authentication in identity between the server and the user, and the server platform is verified also. The proposed scheme meets the designed goals, is security and reliable. Compared with other schemes, it is more secure and more comprehensive in security features, can facilitate the application of the existing trusted network frameworks [16], can supply the users with more security services. While the proposed scheme needs trusted computing environment, then we will further focus on the credibility of the server and general server co-exist in the environment and the credibility of the certification of the problem of building reliable server environment, for examples, we will focus on how to use trusted virtual machine to realize a trusted server and etc.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable suggestions. This work is supported by the Key Program of National Nature Science Foundation of China (60633020), the Fundamental Research Funds for the Central Universities (K50510030003, JY10000903001).

References

[1] H. Y. Chien, J. K. Jan, and Y. M. Tseng, "An efficient, and practical solution to remote authentication: Smart card," *Computers & Security,* vol. 21, no. 4, pp. 372-375, 2002.

[2] L. L. Chun and T. L. Hwang, "A password authentication scheme with secure password updating," *Computers & Security*, vol. 22, pp. 68-72, 2003.

[3] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory,* vol. 22, pp. 644V654, 1976.

[4] P. George, "User authentication wit h smart cards in trusted computing architecture," *Proceedings of the International Conference on Security and Management*, pp. 25-31, Las Vegas, Nevada, USA, 2004.

[5] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, pp. 770-772, 1981.

[6] W. C. Ku and S. M. Chen, "Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards," *IEEE Transaction on Consumer Electronics*, vol. 50, pp. 204-207, 2004.

[7] M. Kumar, "A secure remote user authentication scheme with smart cards". (http://eprint.iacr.org /2008/331)

[8] I. E. Liao, C. C. Lee, and M. S. Hwang, "A password authentication scheme over insecure networks," *Journal of Computer and System Sciences*, vol. 72, pp. 727-740, 2006.

[9] S. Pearson, "Trusted Computing: Strengths, Weaknesses, and Further Opportunities for Enhancing Privacy," *iTrust 2005*, LNCS 3477, pp. 305-320, Springer-Verlag, 2005.

[10] S. Rehbocka and R. Hunt, "Trustworthy clients: Extending TNC to web-based environments," *Computer Communications*, vol. 32, pp. 1006-1013, 2009.

[11] C. X. Shen, H. G. Zhang, D. G. Feng, *et al.*, "Survey of Information Security," *Science China Information Sciences*, vol. 50, pp. 273-298, 2007.

[12] C. X. Shen, H. Zhang, H. M. Wang, *et al.*, "Research on trusted computing and its development," *Science China Information Sciences*, vol. 53, pp. 405-433, 2010.

[13] F. Stumpf, O. Tafreschi, P. Roder, *et al.*, "A Robust integrity Reporting Protocol for Remote Attestation," *Proceedings of the Second Workshop on Advances in Trusted Computing*, Tokyo, Japan, 2006.

[14] Trusted computing group, *TCG Specification Architecture Overview*. (http://www. trustedcomputinggroup.org)
[15] Trusted Computing Group, *TPM Main Specifications - Part 1 Design Principles*. (http://www.trustedcomputinggroup.org)
[16] Trusted Computing Group, *TCG Trusted Network Connect TNC Architecture for Interoperability XSpecification Version 1.2*, 2007. (http://www.trustedcomputinggroup.org)
[17] C. S. Tsai, C. C. Lee, and M. S. Hwang, "Password authentication schemes: Current status and key issues," *International Journal of Network Security*, vol. 3, pp. 101-115, 2006.
[18] G. Yang, D. S. Wong, H. Wang, and X. Deng, "Two-factor mutual authentication based on smart cards and passwords," *Journal of Computer and System Sciences*, vol. 74, pp. 1160-1172, 2008.
[19] E. J. Yoon, K. Y. Yoo, and Drawbacks of Liao *et al.*, "Password Authentication Scheme." *International Conference on Next Generation Web Services Practices*, pp. 25-28, Sep. 2006.

Li Yang received the B.S. degree in instructional technology from Shannxi Normal University, Xian, P. R. China in 1999, and M. S. degree in computer science from Xidian University, in 2005, and the Ph. D. degree in cryptography from Xidian University, Xian, P. R. China, in 2010. Now he is a lecturer in of the Department of Computer Science, Xidian University. His current interests include trusted computing, wireless security, and cryptography.

Jian-feng Ma received the B.S. degree in computer science from Shaanxi Normal University, Xian, P. R. China in 1982, and M. S. degree in computer science from Xidian University, in 1992, and the Ph. D. degree in computer science from Xidian University. Currently he is a Professor of the Department of Computer Science, Xidian University. He has published over 50 international journal and conference papers. His research interests include information security, cryptography, and network security.

Qi Jiang received the B.S. degree in computer science from Shaanxi Normal University, Xian, P. R. China in 2004, and he is currently pursuing the Ph. D. degree in computer science at Xidian University. His current interests include network security and cryptography.