International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(2): 384-395
The Society of Digital Information and Wireless Communications, 2011 (ISSN: 2220-9085)

An Enhanced Remote Authentication Scheme using Secure Key Exchange Protocol with Platform Integrity Attestation

Fazli Bin Mat Nor[1], Kamarularifin Abd Jalil[1], and Jamalul-lail Ab Manan[2],

[1]Faculty of Computer & Mathematical Sciences, Universiti Teknologi Mara, 40450 Shah Alam, Selangor, Malaysia.
[2]MIMOS Berhad, Technology Park Malaysia, 57000 Bukit Jalil, Kuala Lumpur, Malaysia.
fazlimnor@hotmail.com, kamarul@tmsk.uitm.edu.my, jamalul.lail@mimos.my

ABSTRACT

Most remote authentication schemes use key exchange protocol to provide secure communication over an untrusted network. The protocol enables remote client and host to authenticate each other and communicate securely with prearranged shared secret key or server secret key. Many remote services environment such as online banking and electronic commerce are dependent on remote authentication schemes to validate user legitimacy in order to fulfill the authentication process. Unfortunately, these schemes are not able to provide trust or evidence of claimed platform identity. Therefore, these schemes are vulnerable to malicious software attacks that could compromise the integrity of the platform used for the communication. As a result, user identity or shared secret key potentially can be exposed. In this paper, we present a remote authentication scheme using secure key exchange protocol with hardware based attestation to resist malicious software attack. In addition, a pseudonym identity enhancement is integrated into the scheme in order to improve user identity privacy.

KEYWORDS

Remote user authentication; remote attestation; trusted platform module; privacy; pseudonym

1 INTRODUCTION

Most remote services such as online banking and e-commerce employ remote authentication to verify user legitimacy in order to for the user to gain access to their services. Remote authentication normally requires user credential to be sent across network in order for another party to validate its authenticity. However, insecure network and improper key exchange mechanism might expose sensitive information such as user credential to attacks like man-in-the-middle and malicious software attack. To address this shortcoming, verifier-based remote user authentication has been introduced by Lamport [1] in order to prevent sensitive information from being exposed over insecure communication between user and the services. Many past works [2],[3],[4],[5] have been proposed to improve remote authentication scheme. However, these improvements are more related to strengthen the key exchange protocol and none of them concentrate on securing client's platform. Normally without any protection, platforms used in the communication are always vulnerable to attacks such as malicious software attacks. These attacks might lead to security instability of the platform or machine. Consequently, user credential or server secret key can potentially be stolen if platform or system used is compromised.

Therefore, an additional security mechanism needs to be introduced in order to detect any illegitimate changes to platform configuration. This security mechanism is important for making sure that platforms used in secure communication are untainted by any malicious software. Malicious software could bring malicious payload that would create or execute unwanted activities without owner's consent. As a result, the unwanted activities might open a pathway to expose sensitive information such as user credential or shared secret key. Therefore, platform integrity must be verified before remote user and host exchanging their sensitive information even in secure communication. Platform integrity measurement and reporting provided by Trusted Platform Module (TPM) [6] are two main features in order to ensure platform integrity is trusted. By having these features in place, one can send their platform integrity information to be verified by other party and this process is called remote attestation. On the other hand, key exchange protocol will be more secure if the key never being sent across the network. As such, Secure Remote Password (SRP) [7] is an ideal key exchange protocol for this requirement.

1.1 Secure Remote Password (SRP) Protocol

Secure Remote Password (SRP) protocol is password authentication and key exchange protocol over an untrusted network and it has been developed based on zero knowledge proof and verifier based mechanism [7]. In the event of authentication, zero knowledge proof allows one party to prove themselves to another without revealing any authentication information such as password. On the other hand, verifier based mechanism requires only verifier that has been derived from password to be stored in the server side. Thus, this protocol makes sure no sensitive authentication information such as password to be sent across the network.
The SRP protocol as shown in Figure 1 consists of two stages. First stage of the protocol is to set up authentication information of the client and store the information on the server side. At this stage, client calculates secret information sent by the verifier based on client's password and random salt. Server then stores client's username (i), verifier (v) and random salt (s) for authentication purposes. Second stage is the authentication process. Steps of SRP authentication are follows [2]:

1. Client then generates a random number (a) and by using generator (g), client calculates public key (A) = ga. Client starts the authentication process by sending public key, A with its username (i) to the server.

2. Server looks up for client's verifier (v) and salt (s) based on the username (i). Server then generates its random number (b) and computes its public key (B) using verifier (v) and generator (g) and server sends (s, B) to client.

3. Upon receiving B and s, client calculates private key (x) based on salt (s) and its password (p).

4. Both client and server then compute their own session key (S) with different calculation method. Session key (S) calculated by both parties will match

when password used in the calculation is originally used to generate the verifier (v).

5. Both sides then generate cryptographically strong session key (k) by hashing session key (S).

6. In order for client to prove to the server that it has correct session key, it calculates M1 and sends to the server. The server verifies the M1 received from client by comparing with its own calculated M1 values.

7. Server then sends M2 to client as evidence that it has correct session key.

8. Finally, once client verifies M2 is matches with its own calculated M2 value, client is now authenticated and secured communication channel can be established.

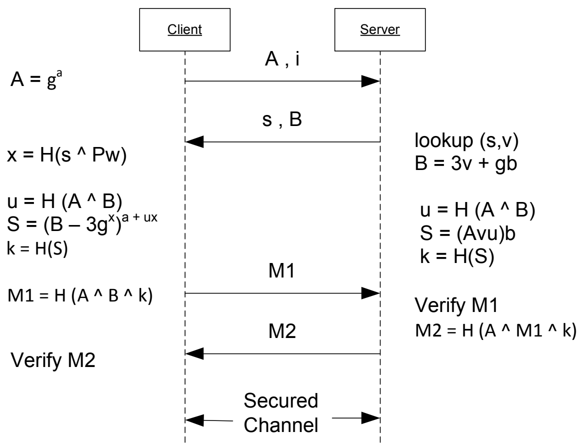

Figure 1. Secure Remote Password authentication protocol

1.2 TPM Based Remote Attestation

Remote attestation allows remote host such as server to verify integrity of another host's (client) platform such as its hardware and software configuration over a network. Thus, by using this method, remote host will be able to prove and trust that client's platform integrity is unaffected by any malicious software. As mentioned by Trusted Computing Group (TCG) [8], an entity is trusted when it always behaves in the expected manner for the intended purpose. Therefore, remote attestation is an important activity to develop trust relationship between client and server to ensure the communication is protected from illegitimate entity.

In remote attestation, client's platform integrity is measured in relation to its hardware and application information and the integrity measurement values will be stored into a non-volatile memory in TPM. Next, the measurement values are integrated as part of integrity report that later will be sent to host system such as server to be analyzed and verified, in order to prove to the host system that its platform integrity is untouched by any unwanted entity.

However, TPM hardware itself is a passive chip. Therefore, TPM alone is unable to measure the platform and it requires software intervention to activate its functionalities. In order to recognize platform as trustworthy platform, platform measurement process has to start at boot time of its host. Trusted boot process such as TrustedGrub [9] measures platform components such as BIOS, boot loader and operating system and extends integrity measurement into 160 bit storage register inside TPM called Platform Configuration Register (PCR) [10, 11]. Therefore, this hardware based integrity measurements can be amalgamated with other application based measurements to produce evidence to other party in attestation process.

Nevertheless, integrity measurement alone cannot provide the identity of the platform. For this reason, each TPM has its unique Endorsement Key (EK) certified by its manufacturer which identifies the TPM identity. To overcome privacy concerns if EK is used directly in attestation process, Attestation Identity Key (AIK) which is derived from the EK is used to sign integrity measurement. Thus, TPM based remote attestation is also crucial to establish the truly trusted identity of the platform to other party.

1.3 Our Contribution

Trust and privacy are important security elements that must be taken care of when dealing with remote services. With this in mind, each parties involve in the communication must ensure that they communicate with legitimate and trusted entities as well as their identity privacy is protected. Thus, it is crucial to incorporate both these elements in authentication scheme related to remote services.

In this paper, we propose remote user authentication protocol that makes use of TPM features to incorporate trust element and protect user's privacy with pseudonym identity. In addition, we also take advantage of SRP key exchange protocol to provide strong session key in the proposed communication.

1.4 Outline

This paper is organized as follows: Section 2 discusses previous works on authentication related to remote services and their issues. Section 3 presents our proposed solution, whereas section 4 and 5 analyze security elements on the proposed protocol. Methodology and experimental result are explained in section 6 and 7. Finally, section 8 concludes the paper.

2 RELATED WORKS

Many of past works on remote authentication protocol [2],[3],[4],[5] have been proposed to overcome insecure communication between client and server. These protocols have solely focused only on user legitimacy and require shared secret key to provide secure communication. However, without any protection to endpoint platform at both client and server, these protocols are still vulnerable to malicious software attack when they reach the endpoints. Furthermore, user credential and shared secret key can be potentially exposed if it is not securely protected at endpoints.

Zhou et al. [12] took initiative to introduce password-based authenticated key exchange and TPM-based attestation in order to have secure communication channels and endpoint platform integrity verification, due to the issue of normal SSL/TLS or IPSec which do not provide endpoint integrity. They proposed Key Exchange with Integrity Attestation (KEIA) protocol which is based on a combination of both password-based authenticated key exchange and TPM-based attestation. This protocol is the first known effort that combines platform integrity to endpoint identity in order to prevent reply attack and collusion attack. KEIA adopts SPEKE [18] as their key exchange protocol. However, Hu [2] stated that SPEKE is susceptible to password guessing attack when simple password is used. On the other hand, KEIA protocol uses prearranged shared secrets as the part of their authentication.

Ali [16] has proposed remote attestation on top of normal SSL/TLS secure channel. His work provides architecture for access control based on the integrity status of the web client. Thus, client with compromised integrity will not be able to access services on the server. However, this solution relies solely on Secure Socket Layer (SSL) for their secure communication. Unfortunately, integrity reporting protocol cannot rely on SSL alone as the protocol is vulnerable to main-in-the-middle attack [12], [13]. Cheng et al. [13] proposed a security enhancement to the integrity reporting protocol by implementing cryptographic technique to protect measurement values. However, their solution requires client to generate premaster secret key and client has to carefully secure the key to avoid impersonation if the key is stolen [7].

3 PROPOSED SOLUTION

In this section, we present remote user authentication protocol with both elements of trust and privacy. For this purpose we decided to use TPM based remote attestation and user identity pseudonymization as trust and privacy implementation method respectively. In addition, SRP is adopted in the proposed protocol as the secured key are being exchanged between communicating parties. Notations used in proposed protocol are describes in Table 1.

Notation	Description
i	User identity (user name)
PCR	Selected PCR value
u	Client pseudonym identification
g	A primitive root modulo *n* (often called a generator). While *n* is a large prime number.
s	A random string used as the user's salt
Pw	The user's password
x	A private key derived from the password and salt
v	Password verifier
a,b	Ephemeral private keys, generated randomly and not publicly revealed, 1 < a or b < n
A,B	Corresponding public keys
H(.)	One-way hash function
m ^ n	The two quantities (strings) m and n concatenated
k	Session key
Mc	Client evidence
Ms	Server evidence
SML_c	Client's Store Measurement Log
SML_s	Known good hashes of Store Measurement Log (Server side)
Sn	Signature value signed with AIK private key
Enc_k	Encryption method with *k* as key
Dec_k	Decryption method with *k* as key

Table 1. Notation of the proposed protocol.

3.1 Protocol Description

Our proposed protocol as shown in Figure 2 consists of two phases; registration phase and verification phase. In registration phase, client sends pseudonym identity, verifier value, random salt value and public certificate of its AIK to the server to set up authentication information via secure channel. Following are the steps for registration process:

1. Client computes its pseudonym identity (u) by hashing combination of user identity and platform PCR values.

2. In order to compute private key (x), client generates random salt value (s) to be combined with client's password in hash function.

3. Client calculates its verifier value (v) derived from private key (x) using generator (g).

4. Client then sends u, v, s and public certificate of its AIK to server. Server then stores that information in database for authentication purposes.

In authentication phase, there are two stages of proof evidence that need to be fulfilled in order to complete the authentication process. First stage, client and server need to proof each other that they are having the same session key (k) without revealing any information about the key. This is done based on zero-knowledge proof calculation implemented in SRP protocol. Second stage, client needs to provide proof to the server that its platform integrity is unaffected by any malicious software. Following are steps for authentication process:

1. Client calculates its pseudonym identity (u) by hashing combination of user identity (i) and selected platform PCR values. Client then calculates its public key (A) using generator (g) and sends both values (A,u) to server.

2. Server looks up client's salt value (s), verifier (v) and public AIK certificate from its database based on pseudonym identity (u) given by client. At the same time, server calculates its public key (B) using generator (g) and sends both values (s, B) to client. Prior to sending the values, server computes its session key (k) based on mathematical calculation stated in SRP protocol.

3. Upon receiving salt (s) and server's public key (B), client computes its session key (k). Client then sends Mc as evidence that it has the correct session key.

4. Once the server verifies Mc is matched with its own calculated Mc, server then computes Ms to prove that it also has the correct session key (k). Server then sends Ms together with random number (nc) to client.

5. Client verifies Ms with its own calculated Ms, if the values matched, client then invokes TPM functionality by signing its platform measurement values stored in PCR with AIK private key. The signature is then encrypted with session key (k) together with hashed values of client's username (i), PCR values and stored measurement log (SMLc). Next, the encrypted values (Ek) are sent to the server.

6. Upon receiving Ek from client, server decrypts the Ek using its session key (k). Server then verifies signature (Sn) with client's AIK certificate. Once the signature is verified, server computes hashed values of pseudonym id (u) with its own stored measurement log (SMLs). Next, server verifies its measurement hashed values with client's measurement hash values. If the verification succeeds, both parties now able to communicate in secured and trusted channel.

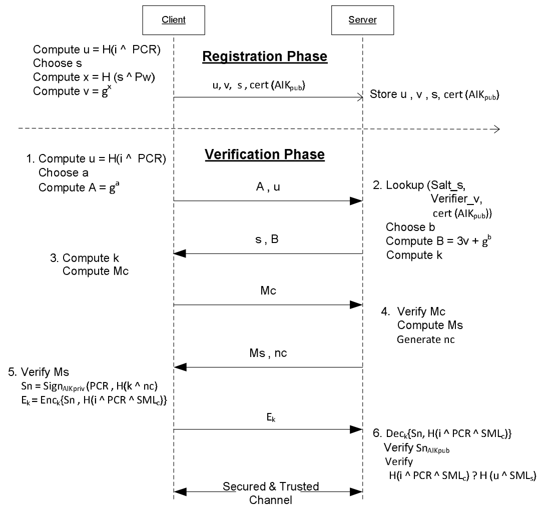

Figure 2. Proposed solution registration and authentication scheme

4 SECURITY ANALYSIS

In this section, we analyze security elements on our proposed solution based on integrity verification, impersonation attack, stolen verifier attack, insider attack and identity protection.

4.1 Integrity Verification

One of the important security elements in our proposed solution is trust. In order to establish trust, client's platform integrity needs to be analyzed by remote host. Therefore, it is crucial to secure client's platform integrity measurement from any illegitimate parties. The proposed assures the integrity measurement is transferred securely. This is done by encrypting the measurement with session key (k). Thus, in order to manipulate the integrity measurement value, attacker would need to capture (k), however it is impossible as (k) has never been exchanged between client and server. Furthermore, our solution uses TPM as tamper-proof hardware that protects all the measurements from being manipulated at client side.

4.2 Impersonation Attack

Attacker would not able to impersonate either client or server without knowing session key (k) as implementation of SRP protocol requires zero knowledge proof. Without session key (k), attacker would not able to compute evidence Mc or Ms, in order to prove he or she has the correct session key. Moreover, session key (k) is never passed over the network and this will make impersonation attack almost impossible.

4.3 Stolen Verifier Attack

Normally when password related information such as verifier is stored at the server side, it is vulnerable to stolen verifier attack. This attack happens when attacker able to gain access to the server and manage to extract verifier information from its database. This attack also might lead to impersonation attack when attacker manages to manipulate authentication process using the stolen verifier. The strength of the proposed is that even though attacker manages to steal the verifier (v), the attacker would not able to continue with authentication process without client's password as it requires expensive dictionary search to reveal it [7].

4.4 Insider Attack

Weak client's password or server secret key stored in server side is vulnerable to any insider who has access to the server. Thus, in the event of this information is exposed, the insider able to impersonate either party. The strength of our proposed protocol is that it does not store any client's password or server secret key in the server side. Therefore, our scheme can prevent the insider from stealing sensitive authentication information.

4.5 Identity Protection

Current remote user authentication protocols [12],[16],[3],[13] lack privacy protection as most of the protocols have no mechanism to protect that information from being linked back to actual user. The proposed preserves user identity privacy by replacing user identity with pseudonym identity (u) which is a hashed value of user identity and selected platform PCR values. Pseudonym identity is important because in the event of server's database has been compromised; user identity privacy is still protected due to the fact that attacker cannot manipulate the pseudonym identity or link it back to actual user.

5 PROTOCOLS COMPARISON

In this section, we summarized the proposed and other related schemes based on security analysis. Table 2 shows comparison between our scheme and other schemes.

Protocols	Security Analysis				
	IV	IA	SV	IT	IP
Our scheme	√	√	√	√	√
Zhou et al. [12]	√	√	√	ø	x
Ali [16]	√	√	√	x	x
Cheng et al. [13]	√	√	√	x	na
Hu et al. [2]	x	√	√	√	√
Liao et al. [3]	x	√	√	x	x
Chien et al. [4]	x	√	√	x	√
Chai et al. [5]	x	√	√	x	√

Table 2. Security analysis summary.

* Notation:
IV - Integrity Verification
IA - Impersonation Attack
SV – Stolen Verifier Attack
IT – Insider Attack
IP – Identity Protection

√ – Satisfied
X – Not satisfied
Ø – Partially satisfied
na – Unrelated

6 METHODOLOGY

Our experimental setup consists of two machines to simulate protocol handshake between client and server. Client machine runs on Intel Core2 Duo @ 2.53GHz with 4GB of RAM and OpenSUSE 11.0 with linux kernel 2.6.30.5 as its operating system. On the other hand, server machine is prepared with Intel Core2 Quad @ 2.4GHz with 8GB of RAM and Ubuntu v9.10 with Linux kernel v2.6.31.22.6 as its operating system. We have setup MySQL v5.5.15 in the server machine as our database storage with default configuration. As our experiment using local area network (LAN), a normal router is used to connect these two machines via network. Network diagram for this implementation is shown in Figure 3.

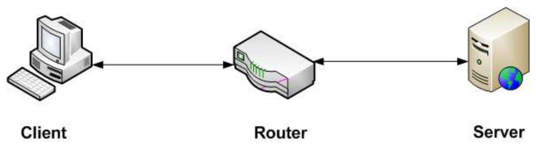

Figure 3. Network diagram

We have chosen public key exchange protocol and SRP key exchange protocols as a performance comparison with the proposed protocol. Implementation of public key protocol with DHE-RSA-AES256-SHA cipher is realized using Transport Layer Security (TLS) 1.0 handshake protocol. As for the TLS implementation, we setup the server machine with Apache web server v2.2.18 [20] with mod_ssl and OpenSSL v0.9.8 [21] as TLS-enabled web server. Implementation of SRP key exchange protocol, we have chosen GnuTLS v2.10.1 [22] as similar representation of TLS handshake protocol. For SRP experimental setup, server machine is equipped with test web server provided by GnuTLS. Setup for our proposed protocol uses customized web server developed in Java. This customized web server is modified from SRPforJava [23] to include TPM remote attestation functionalities. On the client machine, three different customized java console applications were installed to represent client program for each key exchange protocols. The client program is used to do handshake with the web server resides on the server machine.

During the experiment, each key exchange protocols are measured individually with three metrics of performance comparison. The metrics are number of concurrent users, size of transmitted data and client requests per second. Purpose of these metrics is to measure handshake processing time between client and server based on different key exchange protocol. As for number of concurrent users metric, client program runs multiple threads to simulate number of concurrent users and goal of this metric to test simultaneous connections a protocol can handle and server response time as the number of concurrent users increase. Second performance metric in this experiment is size of transmitted data and this metric is chosen to show that handshake duration between client and server is not only dependent on network environment but also size of transmitted data. Thus, goal of this metric is to analyze interaction between protocols and the network environment as well as transmitted data. Last metric in our performance measurement which is client requests per second is to measure protocol and server performance on handling client request whenever number of concurrent user increase.

7 EXPERIMENTAL RESULTS

In order to test the proposed protocol, an experiment was conducted. From the experiment, three different results have been obtained with each result represents specific metric used in the performance comparison.

7.1 Number of Concurrent User

Figure 4 provides an insight into server response time whenever there is increase in number of concurrent user and with different key exchange protocols used. SRP key exchange protocol implemented in GnuTLS achieves better server response time. This is simply because the SRP uses negligible amount of processor time in its protocol operations [7]. The proposed protocol seems slightly higher in term of server response time compared to other protocols. In this case, implementation of TPM based attestation introduces performance bottleneck in the proposed protocol due to TPM needs to perform expensive operation such as TPM_Qoute in order to generate client attestation signature. However, the proposed protocol still comparable with commonly used public key protocol implemented in OpenSSL.

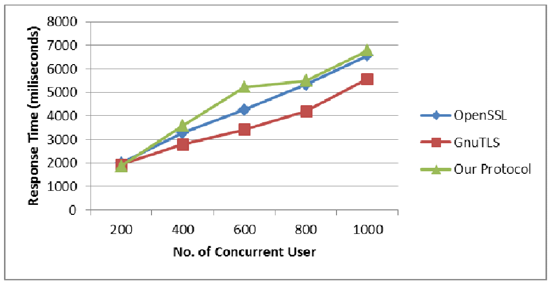

Figure 4. Server response time based on number of concurrent user.

7.2 Size of Transmitted Data

Performance comparison based on size of transmitted data in Figure 5 shows significantly higher processing time for public key exchange. Significant degrade of the OpenSSL performance is due to its public key [19] and also due to the certificate information insertion in the transmitted data. On the other hand, the SRP key exchange protocol reduces size of transmitted data because only key information is exchanged between client and server and does not involve any certificate. Thus, GnuTLS and the proposed protocol which uses SRP as its key exchange protocol perform better in terms of response time. However, the proposed protocol requires platform attestation information to be exchanged during handshake process. As a result, the attestation information increases size of transmitted data and eventually the response time is slightly higher than GnuTLS.

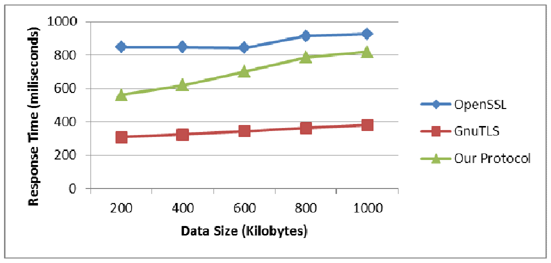

Figure 5. Server response time based on size of transmitted data.

7.3 Client Request per Second

Figure 6 shows that for the GnuTLS implementation the number of client request that the server handles per second is more compared to the others. This is due to the SRP key exchange protocol which requires smaller size of transmitted data and no hardware intervention required. As for the TPM, despite of using similar SRP key exchange as GnuTLS, TPM based attestation included in the proposed seems to have impact on client and server handshake performance. Similarly, OpenSSL also suffers with performance issue when number of concurrent users increase due to certificate information used in handshake process which forced the client and server to process larger size of transmitted data.

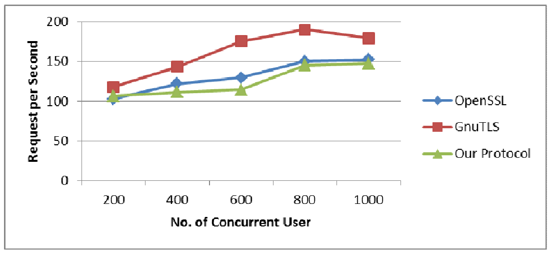

Figure 6. Number of client request per second handle by server based on number of concurrent user.

8 CONCLUSION

In this paper, we have shown that the current remote user authentication schemes require some improvement in terms of providing protection from malicious software attack and preserving user identity privacy. We propose trusted and secure remote user authentication with privacy enhancement to user identity in order to fulfill limitation of current schemes. The proposed solution incorporates TPM based attestation and SRP key exchange protocol to provide trusted and secure communication between client and server. In addition, the proposed protocol preserves user identity privacy by replacing actual user identity with pseudonym identity. We demonstrate security analysis on proposed protocol based on a few security criteria which shows that the proposed protocol resists any possible threats. We also conducted a performance analysis in order to show smaller size of transmitted data is required in order to increase performance of protocol handshake. Performance analysis also shows that the platform based attestation does give impact to client and server processing time but it is still comparable with commonly use TLS protocol implementation such as OpenSSL.

9 ACKNOWLEDGEMENT

The authors would like to thank Universiti Teknologi MARA for funding this research under the Excellence Fund.

10 REFERENCES

1. Lamport, L.: Password authentication with insecure communication. Communications of the ACM 24(11), pp. 770–772 (1981)
2. Hu, L., Yang, Y., Niu, X.: Improved remote user authentication scheme preserving user anonymity. In: Communication Networks and Services Research (CNSR '07), pp. 323-328 (2007)
3. Liao, Y.P., Wang, S-S.: A secure dynamic ID based remote user authentication scheme for multi-server environment. In: Computer Standards & Interfaces, 31(1), pp. 24-29 (2009)
4. Chien, H.-Y., Chen, C.-H.: A remote authentication scheme preserving user anonymity. In: Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA '05), v 2, Washington, USA, pp. 245-248 (2005)
5. Chai, Z., Cao, Z., Lu, R.: Efficient password-based authentication and key exchange scheme preserving user privacy. In: International conference on Wireless Algorithms, Systems, and Applications (WASA), pp. 467-477 (2006)
6. Sadeghi, A.: Trusted Computing – Special aspects and challenges. In: SOFSEM 2008. LNCS, vol. 4910, pp.98-117. Springer, Slovakia (2008)
7. Wu, T.: The Secure Remote Password protocol. In: Internet Society Network and Distributed Systems Security Symposium (NDSS). San Diego, pp. 97-111 (1998)
8. Trusted Computing Group: TCG specification architecture overview, specification revision 1.4 (2007)
9. TrustedGrub, http://www.sirrix.com/content/pages/trustedgrub.htm
10. Kinney, S.: Trusted Platform Module Basics: Using TPM in Embedded System. NEWNES (2006)
11. Challener, D., Yoder, K., Catherman, R., Safford, D., Doorn, L.V.: A Practical Guide to Trusted Computing. IBM Press (2008)
12. Zhou, L., Zhang, Z.: Trusted channels with password-based authentication and TPM-based attestation. In: International Conference on Communications and Mobile Computing, pp.223-227 (2010)
13. Cheng, S., Bing, L., Yang, X., Yixian, Y., Zhongxian, L., Han, Y.: A security-enhanced remote platform integrity attestation scheme. In: Wireless Communications, Networking and Mobile Computing (WiCom '09), pp. 1-4, 24-26 (2009)
14. Juang, W.S. , Wu, J.L.: Efficient user authentication and key agreement with user privacy protection. In: International Journal of Network Security. v7, pp. 120-129 (2008)
15. Zhang, M.: Analysis of the SPEKE password-authenticated key exchange protocol. In: IEEE Communications Letters 8(1), pp. 63-65 (2004)
16. Ali, T.: Incorporating remote attestation for end-to-end protection in web communication paradigm. In: International Conference on Internet Technologies and Applications. Wales, UK (2009).
17. Stumpf, F., Tafreschi O., Röder, P., Eckert, C.: A robust integrity reporting protocol for

remote attestation. In: Proceedings of the Workshop on Advances in Trusted Computing (WATC) (2006)

18. Jablon, D.: Strong password-only authenticated key exchange. In: SIGCOMM Computing Communication 26(5) (1996)
19. G. Apostolopoulos, V. G. J. Peris, and D. Saha: Transport Layer Security: How Much Does It Really Cost?. In: INFOCOM, pp. 717–725 (1999)
20. Apache Web Server, http://httpd.apache.org/
21. OpenSSL, http://www.openssl.org/
22. GnuTLS, http://www.gnu.org/software/gnutls/
23. SRPforJava, http://code.google.com/p/srpforjava/