International Journal of Cyber-Security and Digital Forensics (IJCSDF) 1(3): 204-210
The Society of Digital Information and Wireless Communications (SDIWC) 2012 (ISSN: 2305-0012)

Mitigating Man-In-The-Browser Attacks with Hardware-based Authentication Scheme

Fazli Bin Mat Nor[1], Kamarularifin Abd Jalil[2]
Faculty of Computer & Mathematical Sciences, Universiti Teknologi Mara, 40450 Shah Alam, Selangor, Malaysia
fazlimnor@hotmail.com[1], kamarul@tmsk.uitm.edu.my[2]

Jamalul-lail Ab Manan
MIMOS Berhad, Technology Park Malaysia, 57000 Bukit Jalil, Kuala Lumpur, Malaysia
jamalul.lail@mimos.my

***Abstract*— Lack of security awareness amongst end users when dealing with online banking and electronic commerce leave many client side application vulnerabilities open. Thus, this is enables attackers to exploit the vulnerabilities and launch client-side attacks such as man-in-the-browser attack. The attack is designed to manipulate sensitive information via client's application such as internet browser by taking advantage of the browser's extension vulnerabilities. This attack exists due to lack of preventive measurement to detect any malicious changes on the client side platform. Therefore, in this paper we are proposing an enhanced remote authentication protocol with hardware based attestation and pseudonym identity enhancement to mitigate man-in-the-browser attacks as well as improving user identity privacy.**

Keywords- Trusted platform module; man-in-the-middle; man-in-the-browser; remote user authentication; privacy; pseudonym

I. INTRODUCTION

Lately, client side attacks on online banking and electronic commerce are on the rise due to inadequate security awareness amongst end users. As a result, end user would not aware if there is vulnerability on their machine or platform that might lead to client side attack such as man-in-the-browser (MitB) attacks. Furthermore, traditional security mechanisms such as antivirus are not efficient enough in preventing these attacks due to it evolves to be more complex with time. For instance, man-in-the-middle (MitM) attack techniques which are mainly targeting the information flow between a client and a server have now evolved to become man-in-the-browser (MitB) attack. MitB attack is designed to infiltrate the client software such as the internet browser and manipulate or steal any sensitive information.

Typically internet browser allows external browser extension to be installed in order to add additional features to the browser. This extension allows user to have extended functionalities other than common browser function. However, by allowing external extension to be installed with or without the user's consent into the end user machine through the browser make it more vulnerable to MitB attacks. In fact, attackers normally use the same distribution channel as legitimate extension to distribute the malicious plugins into the user's browser. Furthermore, with lack of knowledge in security awareness, end user would not be able to differentiate the genuine or malicious extension. Consequently, malicious plugins are able to infiltrate into the browser and would be able to manipulate any sensitive information between a client and a browser.

Therefore, in order to mitigate this issue, the integrity of the client software as well as the user's platform used in the communication must be validated to detect any illegitimate changes to the platform or application. Furthermore, to protect sensitive information such as user credential from being tampered or manipulate, it must be securely transmitted between client and server. For this reason, a hardware-based remote attestation is chosen to provide platform integrity checking requirement in our proposed protocol. Specifically, Trusted Platform Module (TPM) [1] based attestation is integrated into our protocol in order to provide trust relationship between interacting platforms. On the other hand, Secure Remote Password (SRP) [21] is chosen as our password authentication and key exchange protocol to authenticate user and securely transmit sensitive information based on zero knowledge proof and verifier based mechanism. In addition, we have applied pseudonym technique to protect and improve the privacy of user identity.

A. *Secure Remote Password*

Due to lack of security awareness amongst the end users, their credentials such as passwords are exposed to attacks such as brute force attack and dictionary based attacks. Typically this situation happens due to weak passwords as it's a weakest link in the internet security. Once adversary manages to get the password, they would be able to gain access to other sensitive information. Thus, Secure Remote Password (SRP) protocol is specifically designed to mitigate this issue. SRP does not require strong passwords to be in place in order to achieve strong security as it has been developed based on zero knowledge proof and verifier based mechanism [21]. In the event of authentication, zero knowledge proof is a method for one party to provide authentication evidence or prove to other party without revealing sensitive authentication information such as password. On the other hand, verifier based mechanism requires only verifier value that derived from the password to be stored at the server side. Both techniques make sure the password is not sent across the network or being stored at the server side.

SRP protocol establishes its authentication process with client calculates private key value (x) based on client's password (P) and random salt (s). Client's verifier (v) then is derived from the private key. Server then stores client's username (i), verifier (v) and random salt (s) for authentication purposes. Steps of SRP authentication are follows [2]:

1. Client sends 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 sends salt (s) to client. Upon receiving s, client computes private key (x) based on (s) and client's password (P) and sends it to the server.

3. Client computes public key (A) based on generated random number (a) and sends it to the server

4. At the same time, server computes its public key (B) based on client's verifier and generated random number (b). Server then sends (B) and randomly generated number (u) to the client.

5. Client and server then compute the session key (S) based on the values available with each other.

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

7. Client then sends M1 as evidence to prove that it has correct session key by hashing value of (A), (B) and (k). The server verifies the M1 received from client by comparing with its own calculated M1 values.

8. Server then sends M2 to client as evidence to prove that it also has the correct session key. Once client verifies the M2 values matches with its own calculated M2 values, both parties now able to use (k) as a session key for their secure communication.

	Client		Server
1.		i →	(lookup s, v)
2.	x = H(s, P)	← s	
3.	$A = g^a$	A →	
4.		← B, u	$B = v + g^b$
5.	$S = (B - g^x)^{(a + ux)}$		$S = (Av^u)^b$
6.	K = H(S)		K = H(S)
7.	M1= H(A, B, K)	M1 →	(verify M1)
8.	(verify M2)	← M2	M2 = H(A, M1, K)

Figure 1 : Secure Remote Password authentication protocol

B. Pseudonym

Expansion of the internet usage is leading to more privacy threats such as disclosure of personally identifiable information due to many users do not realize they are submitting or exposing sensitive information that could be linked back to their real life identity. In a nutshell, privacy is an ability of the user to be able to protect his or her personal identifiable information and anonymity is often use as a protection of the privacy [22]. Most of authentication schemes normally require user identity and this information is kept inside the server side for the future authentication purposes. However, if the server compromised, the user identity information might be compromised as well. Thus, it is important to conceal the information and pseudonym is one of anonymity methods to mitigate this situation.

Pseudonym in authentication scheme is to make sure concealed user identity cannot be linked back to the original identity. Therefore, in order to strengthen the pseudonym identifier and provide unlinkability, it is suggested the identifier is processed from the combination of user identity with other identity such as platform identity. This method is to make sure actual user identity is not stored on the server side and the server only stores pseudonym identifier for the authentication purposes.

C. Man-in-the-Browser (MitB) Attack

As many security measures have been implemented to prevent MitM attacks such as Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocol, adversaries have come out with a new variant of MitM attack which is known as the Man-in-the-Browser (MitB) attack. Similar to the MitM, the MitB trojans such as Zeus/SpyEye, URLzone, Silent Banker, Sinowal and Gozi [2] are used to manipulate the information between a user and a browser and is much harder to detect due to its nature of attacks [2, 3]. This is apparent when an adversary secretly attaches its malicious code into a browser extension or plugins that seems to be doing legitimate activities.

According to Nattakant [4], a browser extension is a small application running on top of the browser and provides extra features to the browser. In fact, the richness and the flexibility of the browser extension or plugins nowadays have somewhat lured malicious software into the user's browser [5]. Thus, the adversaries are taking advantage of the browser extension features and its simple deployment in order for them to implement the MitB attack.

The chronology of a MitB attack is shown in figure 2 and the explanation is given below:

- Once the malicious browser extension infects the user's browser, it sits inside the browser and waits for the user to visit certain websites which are related to the online banking or electronic commerce.
- When the malicious extension found some related patterns such as transaction details while scanning through the user visited websites, it logs any information entered by the user such as credentials.

Furthermore, the malicious extension can even manipulate the transaction information before the user sends it to the server.

- For example, when a user try to do a transaction of $100 to bank account A, the malicious extension will then change the amount information to $1000 as well as changing the recipient bank account to B. The server will not be able to differentiate between the original and the manipulated transaction because the information come from a legitimate user.

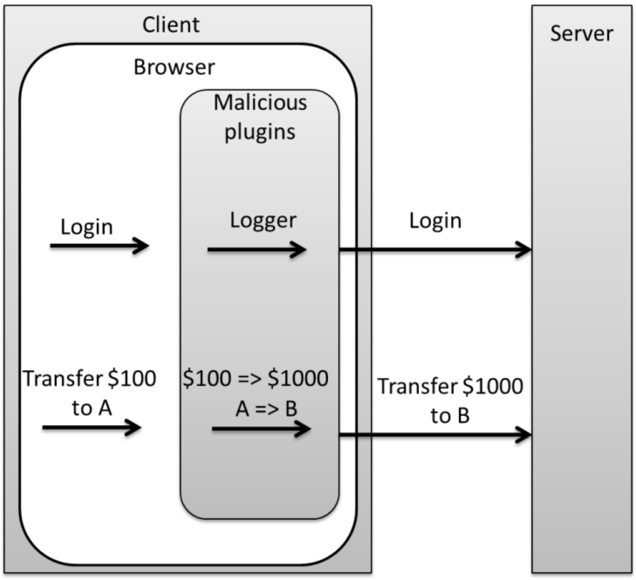


Figure 2 : MitB Attacks

D. Trusted Platform Module Based Remote Attestation

Trust relationship between interacting platforms (machines) has become a major element in increasing the user confidence when dealing with Internet transactions especially in online banking and electronic commerce. For instance, users might want to make sure that they are dealing with a legitimate merchant and vice versa. However, how do you make sure that your browser and machine are trusted and behave in the expected manner while doing the Internet transactions? For this reason, we have chosen the Trusted Platform Module (TPM) due to its capability to provide attestation based on the information about the platform and also can ensure the integrity of the platform is not tampered [1]. In order to ensure the validity of the integrity measurement from the genuine TPM, the Attestation Identity Key (AIK) is used to sign the integrity measurement. AIK is an asymmetric key and is derived from the unique Endorsement Key (EK) certified by its manufacturer which can identify the TPM identity.

As mentioned by the Trusted Computing Group (TCG) [6], an entity is considered trusted when it always behaves in the expected manner for the intended purpose. Due to this reason, the attestation based information provided by the TPM must be verified by the communicating parties before any transaction or sensitive information is transferred. Therefore, the remote attestation is the best option because it allows a remote host such as a server to verify the integrity of another host's (client) platform.

In the remote attestation as shown in figure 3, a client platform will start the attestation process by sending the request for the service. Then, the host platform will send a challenge respond. The client platform will then measures its integrity information such as its operating system, BIOS, hardware and software and this information will be stored in a non-volatile memory in the TPM known as the Platform Configuration Register (PCR) [7, 8]. The client will then send the integrity report to the host for verification. The host will allow the client to access its services once the client's platform integrity has been verified.

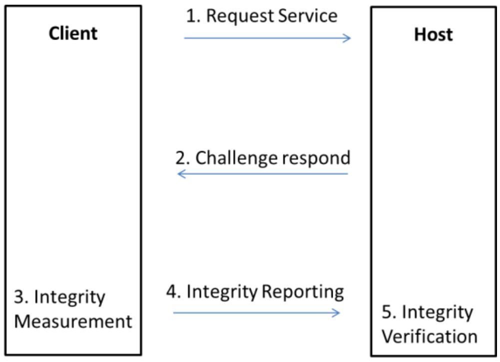


Figure 3: Remote Attestation

E. Our Contribution

The platform integrity and the secrecy of a user's sensitive information are essential to combat MitB attacks. It is more evident since the existing security measures such as antivirus are not capable to prevent new and sophisticated attacks from malicious browser extension [**Error! Reference source not found.**]. Thus, it is crucial to develop trust relationship between a client and a server in order to ensure that the communication is protected from the illegitimate entity as well as to preserve the secrecy of the user's confidential information.

Therefore, in this paper we are proposing a TPM based remote attestation in order to provide trust relationship between the communicating parties. In addition, we have also incorporated pseudonym technique with the secure key exchange in order to prevent the user's confidential information from being tapped. Thus, our proposed solution should able to give better confidence level of users who use browsers to do internet transactions such as on-line banking, on line services.

F. Outline

This paper is organized as follows; section 2 discusses the related works on MitB attacks and its solutions. In section 3, we present our proposed solution; while section 4 discusses the security analysis on the proposed protocol. Finally, section 5 concludes the paper.

II. RELATED WORKS

In the past, several protection mechanisms have been proposed to prevent man-in-the-browser attacks. Bottani et al. [10] has proposed to use personal trusted devices such as mobile phone to secure the transactions between a client and a merchant. This mechanism deters MitB attacks, by avoiding important information such as payment to be submitted through external devices. The IBM research team [11] has also introduced external device approach to prevent MitM variant attacks called Zurich Trusted Information Channel (ZTIC). The ZTIC is an USB device containing components such as simple verification display and other authentication components in order to provide secure communication between a client and a server. In this approach, ZTIC acts as a main medium between the client and the server without relying on any client's application or browser. ZTIC will scan and intercept any sensitive information and will only permit to exchange the information once the client has verified the information within its display component. However, the external devices requirement in the above solutions may become barrier for some users.

On the other hand, Itoi et al. [12] has introduced an Internet based smartcard and this solution uses Simple Password Exponential Key Exchange (SPEKE) to address the security issues regarding the off-line dictionary attack and man-in-the-middle attack. However, this mechanism is unable to provide a protection or notification whenever the platform has been exposed to malicious software. Based on that challenge, Starnberger et al. [13] took the initiative to introduce a smartcard-based TPM attestation by integrating the platform integrity verification in their proposed solution. Thus, it is able to mitigate malicious software attacks on the client's machine as well as providing extra security measurement through external devices.

Abbasi et al. [14] has proposed a secure web contents to mitigate the MitM and the MitB attacks. In their solutions, the web contents from a server will be encrypted by a secured web server and a secured proxy on the client side will decrypt the web content. Thus, this solution provides better protection of web content stored in the server. However, this solution concentrates only on the protection of a web contents on the server side and is lacking of protection against the malicious software attack on the client machine.

Sidheeq et al. [15] has integrated the biometrics USB with the TPM in order to mitigate the risk of malware attacks on the client's machine. In detail, this solution requires the user to provide biometrics evidence for authentication. Then it compares the authenticity of the provided evidence with the user's biometrics stored in the USB which is protected by the TPM. In addition, this solution uses the encryption feature provided by the TPM in order to secure the data exchanged between a client and a server.

III. PROPOSED SOLUTION

In this section, we present our proposed solution to mitigate the MitB attacks (refer to Figure 4). The objective of our protocol is to provide trust communication between a client and a server as well as preserving the secrecy of the user's sensitive information. In order to achieve this objective, we have incorporated the TPM based remote attestation in order to provide the platform integrity verification. In addition, we have adopted the Secure Remote Password (SRP) [16] as the secure key exchange protocol in order to provide zero knowledge proof that allows one party to prove themselves to another without revealing any authentication information such as password.

The proposed protocol comprises of the registration process, the key exchange and also the platform attestation phases. However, before these phases can take place, a client will measure its platform integrity information such as its BIOS, bootloader, operating system and software and store it in the PCR values. The PCR values will be used in the registration as part of the pseudonym data in order to preserve the secrecy of the user's confidential information and also in the attestation process as part of its integrity reporting.

In the registration phase, the client will generate a pseudonym identity (u) by hashing the combination of the user identity and the platform PCR values. The client then calculates the verifier (v) value by hashing the random salt value (s) with the client's password. Next, the client sends (u), (v), (s) and the public certificate of its AIK (a) to the server via a secured channel. The server then stores that information in the database for the authentication purposes.

In the authentication phase, in order to fulfill zero knowledge proof requirements, each party is required to show their evidence that they are using the same secure key exchange without revealing their key. The client will start the authentication process by calculating its asymmetric key and sends the public key (A) and pseudonym identity (u) to the server. The server then lookup the client's (v), (s) and (a) from its database based on the (u) given by the client. Subsequently, the server will calculate its asymmetric key and send the public key (B) and (s) to the client. Prior sending (B) and (s) to the client, the server will calculate its key exchange (k). Upon receiving (s) and also the server's public key (B), the client computes its key exchange (k). The client then sends M1 as evidence to the server. The M1 is calculated based on the mathematical formula stated in the SRP protocol and M1 is used by the server to verify that the client has the same key exchange. Once M1 has been verified, the server then sends its evidence, M2, to the client and the client verifies M2 evidence in order to make sure that the server has the same key. By using this method, both parties will be able to prove to each other that they have the same secure key without revealing the key.

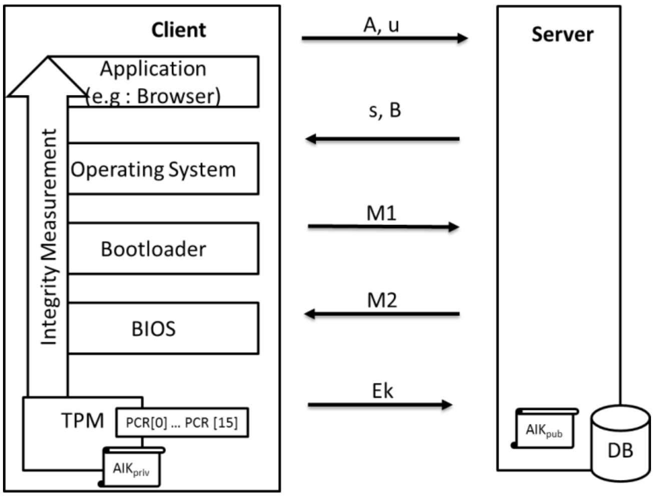


Figure 4: Key Exchange and Attestation Phase

In the platform integrity attestation phase, the client starts the attestation process by signing its PCR values that contains its platform measurements with the AIK private key. The client then encrypts the signature with (k) as the secret key. The encrypted value (Ek) is sent to the server for the integrity verification. Upon receiving (Ek), the server will decrypt it using its secure key (k) and will verify the client's PCR signature with the client's AIK public key (a). Once verified, the client and the server are now able to communicate in a trusted and secure channel.

IV. EXPERIMENTS

In order to test the proposed protocol, we have conducted several experiments. The setup for the experiments consists of three machines: a client machine, a server machine installed with database storage and an adversary machine. The client machine is integrated with the TPM and runs on the Intel Core2 Duo @ 2.53GHz with 4GB of RAM and Ubuntu v9.10 with Linux kernel v2.6.31.22.6 as its operating system. On the other hand, the 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. Database storage in the server machine is configured with MySQL v5.5.15 with default configuration. On the client side, we have prepared a custom internet browser in order to fulfill our proposed protocol requirements. As for the adversary machine, it is equipped with Ubuntu v9.10 with Linux kernel v2.6.31.22.6 and running on Intel Core i7 @ 2.8GHz with 4GB of RAM.

In our first experiment, we simulated the transaction between the untrusted client machine and the server whereby the adversary is furnished with the legitimate user credential and untrusted internet browser. In the second experiment, we have setup man in the middle attack between the trusted client and the server as shown in Figure 5 with the Ettercap v0.7.3 [17] installed at the adversary machine.

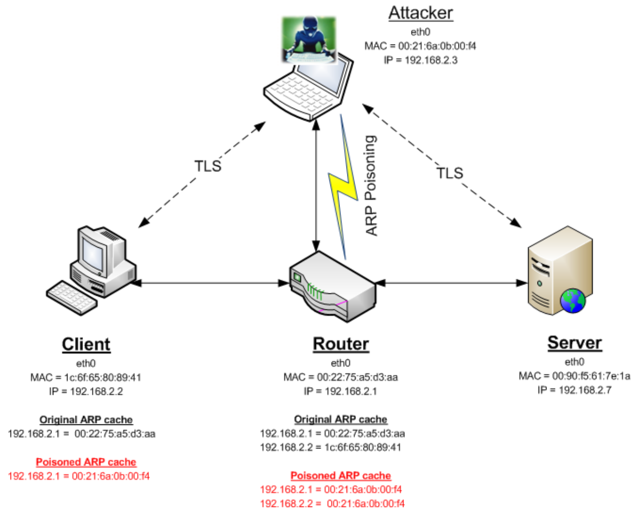
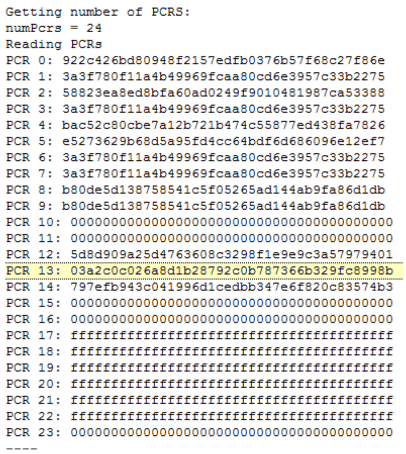


Figure 5: Man in the middle attacks

In our last experiment, we planted a MitB trojan, Zeus [18] into the trusted client machine and have tried to manipulate the transaction content between the client and the server. We specifically measured the integrity of the client operating system related files and applications such as the Internet browser and stored it in the PCR-13 as shown in Figure 6 and this PCR value is used to ensure that the integrity of the client platform and application is not tampered.



```
Getting number of PCRS:
numPcrs = 24
Reading PCRs
PCR 0: 922c426bd80948f2157edfb0376b57f68c27f86e
PCR 1: 3a3f780f11a4b49969fcaa80cd6e3957c33b2275
PCR 2: 58823ea8ed8bfa60ad0249f9010481987ca53388
PCR 3: 3a3f780f11a4b49969fcaa80cd6e3957c33b2275
PCR 4: bac52c80cbe7a12b721b474c55877ed438fa7826
PCR 5: e5273629b68d5a95fd4cc64bdf6d686096e12ef7
PCR 6: 3a3f780f11a4b49969fcaa80cd6e3957c33b2275
PCR 7: 3a3f780f11a4b49969fcaa80cd6e3957c33b2275
PCR 8: b80de5d138758541c5f05265ad144ab9fa86d1db
PCR 9: b80de5d138758541c5f05265ad144ab9fa86d1db
PCR 10: 0000000000000000000000000000000000000000
PCR 11: 0000000000000000000000000000000000000000
PCR 12: 5d8d909a25d4763608c3298f1e9e9c3a57979401
PCR 13: 03a2c0c026a8d1b28792c0b787366b329fc8998b
PCR 14: 797efb943c041996d1cedbb347e6f820c83574b3
PCR 15: 0000000000000000000000000000000000000000
PCR 16: 0000000000000000000000000000000000000000
PCR 17: ffffffffffffffffffffffffffffffffffffffff
PCR 18: ffffffffffffffffffffffffffffffffffffffff
PCR 19: ffffffffffffffffffffffffffffffffffffffff
PCR 20: ffffffffffffffffffffffffffffffffffffffff
PCR 21: ffffffffffffffffffffffffffffffffffffffff
PCR 22: ffffffffffffffffffffffffffffffffffffffff
PCR 23: 0000000000000000000000000000000000000000
----
```

Figure 6: PCR values with trusted client machine

V. RESULTS

Our first experimental result indicates that the adversary's machine would not be able to do any transactions with the server as shown in Figure 7, even though the simulated transaction has been equipped with the legitimate user credential as shown in Figure 8 and Figure 9.

Figure 7: Untrusted client access

This is due to the lack of valid PCR values at the adversary's machine and thus not able to provide the requested valid identity (u) which is the combination of the user identity and the platform PCR values.

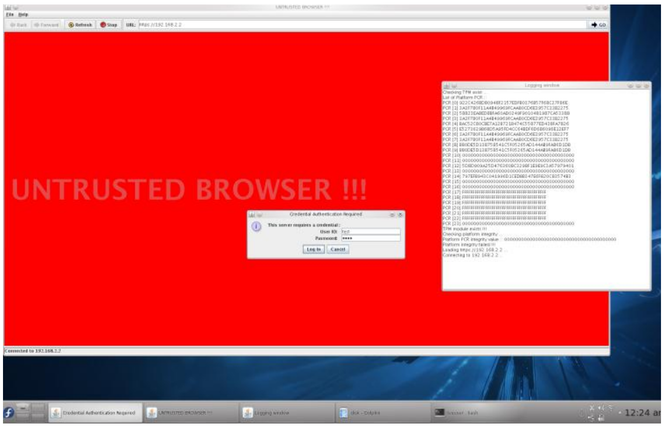


Figure 8: Tampered client application

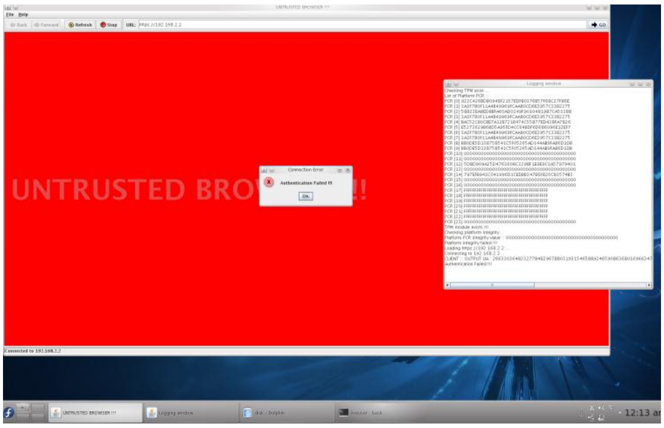


Figure 9: Authentication fail on tampered client application

In the second experiment, the adversary tried to impersonate as a server to the client and as a client to the server. However, the experiment result is similar with our first experiment and it shows that the adversary would not be able to impersonate either as a client or a server due to the invalid evidence (M1) and (M2) in order to fulfill the zero knowledge proof. Furthermore, the platform integrity measurement provided by the adversary's machine is invalid. On the other hand, the adversary would not be able to steal any information from the transaction as the secret key (k) never being exposed in the transaction.

Our last experimental result shows that the PCR-13 value in the trusted client machine has changed as shown in Figure 10. This indicated that the integrity of the client machine has been tampered and in this scenario, the MitB trojan, Zeus has tampered the integrity of the client platform and applications. Consequently, our protocol will reject any authentication with invalid PCR values and Zeus would not be able to manipulate any transaction information.

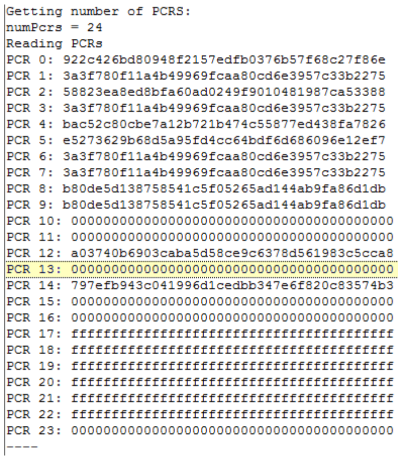


Figure 10: PCR values with untrusted client machine

VI. SECURITY ANALYSIS

In this section, we analyze the proposed protocol based on a few vulnerabilities that could compromise the security of the protocol. According to Oppliger [5], there are three potential vulnerabilities that are related to the client-side attacks: credential-stealing attack, channel-breaking attack and content-manipulation attack.

In the credential-stealing attacks, the adversaries normally use the offline medium such as malicious attachment via email, fake website or other phishing techniques to persuade user to expose their credential unintentionally [5]. In order to mitigate the risk of this attack, the platform integrity must be verified free from any malicious software and the credential must not be transferred or exposed through communication channel. Therefore, the TPM based attestation implemented in our protocol should be able to provide verification of platform integrity. In addition, the credential-free authentication can be achieved through SRP. As a result, when malicious software is planted and embedded into a client's browser machine, the server will automatically reject the client's authentication due to the detection (i.e. different measured integrity value compared to the one in the PCR) of the compromised client's machine in the attestation phase. This capability of detecting phishing attacks and preventing them will certainly increase user confidence in using the internet for transactional activities.

In the channel-breaking attacks, generally the adversaries will act as the man in the middle between a client and a server. In this attack, the adversaries will act as a server to the client and as a client to the server by maintaining a legitimate secure connection between both sides. Thus, any information exchanged between the client and the server will be routed to the adversaries. However, with the adoption of zero knowledge proof of the SRP protocol, the adversaries would need to produce a secure key (k) in order to imitate as a client or a server. Regrettably for the adversaries, in our protocol, the secure key (k) is never exposed or sent across the network. Therefore, the adversaries would not be able to provide correct evidences (M1 or M2) that would enable them to try to get any kind of benefit from the internet transactions.

In the content-manipulation attacks, the adversaries would try to manipulate the content or the transaction information on the client-side i.e. it will tamper the content or the transaction information that is sent to the server. The use of integrity measurement in the remote attestation of the client platform and the browser application becomes the main objective to mitigate this attack. In this case, any content manipulation is strictly detected and prevented by our solution. Therefore, the TPM based integrity measurement implemented in our protocol is able to fulfill the objective.

VII. CONCLUSION AND FUTURE WORKS

In this paper, we have proposed an enhanced remote authentication scheme that mitigates the man-in-the-browser attacks. Briefly, we presented our solution to mitigate such attacks, i.e. the TPM based remote attestation and the SRP key exchanges that form the basis of our protocol which provides the much needed trust, integrity and zero knowledge proof authentications for both the client and the server. In order to test the proposed protocol, we have conducted some experiments. The result obtained from the experiment shows that the proposed protocol is able to deter the attacks launched by the adversaries. We have also demonstrated our security analysis of the proposed protocol based on a few vulnerabilities related to the client-side attacks and we hope that the proposed protocol can effectively deter the attacks. For future works, it is hope that the proposed protocol can be implemented in the common internet browsers such as Firefox in order to have better approach to mitigate MiTB attacks in actual environments. Last but not least, we will look into alternative solutions for strengthening this proposal such as optical tokens, H-PIN and hTAN [19] or behaviour driven security [20]

ACKNOWLEDGMENT

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

REFERENCES

[1] A. Sadeghi, “Trusted Computing – Special aspects and challenges” SOFSEM 2008. LNCS, vol. 4910, pp. 98—117, 2008.

[2] RSA Lab, “Making Sense of Man-in-the-browser Attacks” http://viewer.media.bitpipe.com/1039183786_34/1295277188_16/MITB_WP_0510-RSA.pdf

[3] O. Eisen, “Catching the fraudulent Man-in-the-Middle and Man-in-the-Browser” Network Security, Volume 2010, Issue 4, pp. 11—12, 2010.

[4] N. Utakrit, “Review of Browser Extensions, a Man-in-the-Browser” Proceedings of the 7th Australian Information Security Management Conference, Perth, Western Australia , 2010.

[5] R. Oppliger, R. Rytz, T. Holderegger, “Internet banking: Client-side attacks and protection mechanisms”. Computer 42(6), pp. 27–33, 2009.

[6] Trusted Computing Group, “TCG specification architecture overview, specification revision 1.4”, 2007.

[7] S. Kinney, “Trusted Platform Module Basics: Using TPM in Embedded System”, NEWNES, 2006.

[8] D. Challener, K. Yoder, R. Catherman, D. Safford, L.V. Doorn, “A Practical Guide to Trusted Computing”, IBM Press, 2008.

[9] B. Armin, L. Felix, S. Thomas, “Banksafe Information Stealer Detection Inside the Web Browser”, RAID 2011, pp. 262-280, 2011.

[10] A. Bottoni, G. Dini, “Improving authentication of remote card transactions with mobile personal trusted devices”, Computer Communications 30(8), pp. 1697–1712, (2007.

[11] T. Weigold, T. Kramp, R. Hermann, F. Horing, P. Buhler, M. Baentsch, “The Zurich Trusted Information Channel: An efficient defence against man-in-the-middle and malicious software attacks”, “Proc. TRUST'2008. LNCS”, vol. 4968,pp. 75-91. Springer, 2008.

[12] N. Itoi, T. Fukuzawa, P. Honeyman, “Secure Internet Smartcards”, First International Workshop on Java on Smart Cards: Programming and Security, pp.73-89, 2000.

[13] G. Starnberger, L. Froihofer, K.M. Goeschka, “A generic proxy for secure smart card-enabled web applications”, Web Engineering, 10th Intl. Conf., ICWE 2010, Vienna, Austria, July 5-9, 2010. Proceedings. LNCS, vol 6189, pp. 370-384. Springer, 2010.

[14] A.G. Abbasi, S. Muftic, I. Hotamov, “Web Contents Protection, Secure Execution and Authorized Distribution”, Computing in the Global Information Technology, International Multi-Conference on, pp. 157-162, 2010 Fifth International Multi-conference on Computing in the Global Information Technology, 2010

[15] M. Sidheeq, A. Dehghantanha, G. Kananparan, “Utilizing trusted platform module to mitigate botnet attacks”, Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on , vol., no., pp.245-249, 2010

[16] T. Wu, “The Secure Remote Password protocol”, Internet Society Network and Distributed Systems Security Symposium (NDSS). San Diego, pp. 97-111, 1998

[17] Ettercap, http://ettercap.sourceforge.net/

[18] Symantec, “Zeus: King of the Bots” http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/zeus_king_of_bots.pdf

[19] L. Shujun, S. Ahmad-Reza, S. Roland, “hPIN/hTAN: Low-Cost e-Banking Secure against Untrusted Computers”, Financial Cryptography, 2010

[20] N. Daniel, “Security in Behaviour Driven Authentication for Web Applications”, Master thesis, Department of Computer Science, Electrical and Space Engineering, Jan 2012

[21] T. Wu, “The Secure Remote Password protocol”, Internet Society Network and Distributed Systems Security Symposium (NDSS). San Diego, pp. 97-111, 1998

[22] I. Goldberg, "Privacy-enhancing technologies for the Internet", Compcon '97 Proceedings, pp. 103 - 109, 1997