Platform Property Certificate for Property-based Attestation Model

Nazanin Borhan
Faculty of Computer Science and Information Technology
University Putra Malaysia

Ramlan Mahmod
Faculty of Computer Science and Information Technology
University Putra Malaysia

ABSTRACT

Binary Attestation is currently used in trusted computing environments involving the standard TCG attestation mechanism. However, this mechanism still has deficiencies in terms of flexibility, privacy and scalability. Thus, to overcome these problems, Property-based Attestation has been proposed. Two important issues should be considered in the context of property-based attestation; these include the content of the property and the protocol design. In this study, the researchers proposed platform property certificate, based on the current certificates of the system as the model's property. In addition, a client-server attestation protocol that could apply this particular property is also proposed. In order to show the feasibility of the model, the proposed model was implemented. The results of the implementation showed that the model is efficient to be used to accept and reject valid and invalid inputs. Hence, security aspects listed as privacy, flexibility, scalability and also integrity of the model is checked, while it is crucial to note that it also fulfils the requirements of property-based attestation with TCG standard specifications.

Keywords

Network-level security and protection, Trusted Computing, Public-Private key Authentication.

1. INTRODUCTION

The requirements of Information Technology (IT) have rapidly been changed by today's improvements. In line with this, the needs for confidentiality, authenticity, integrity, anonymity, non-repudiation and availability of the system are becoming more crucial. Security requirements like cryptography, firewalls, and intrusion detection systems are some tools which can be used to ensure the security of a system. Considering the fact that platform trust-ability is the first condition in using these methods, different security problems which result from the weaknesses of software, hardware and their complexity have become more serious.

Trusted Computing Group (TCG) is an available source to improve the new technology called Trusted Computing (TC) that provides a basis to the highest security level in hardware and software. The goal of TCG that is composed of IT infrastructure is to provide a mechanism for the security and integrity of computing platforms. Trusted Platform Module (TPM) and Trusted Software Stack (TSS) are the core components of the TCG which have published some specifications for using them. In more specific, TPM is a tamper resistant hardware that has limited cryptography functionalities. Support of cryptography keys, generation of random numbers in hardware, cryptographically binding data to certain configuration, sealing the data in the configuration of the application, and giving authentication to the platform or application -called Remote Attestation- are some of the capabilities that force many of the cheap seller embedding TPM in their computers [1, 2].

In order to authenticate to a remote party or for a remote party to verify the authenticity of the application, a system uses an approach known as Remote Attestation [3, 4]. There are different methods for attestation. Binary attestation, as it is described in the TCG specification is the standard TCG approach for attestation. Despite the advantages of using the TCG binary attestation, it has its own deficiencies, as described in the following:

First: Flexibility, because data that bound to a particular configuration are considered to be unreachable after system migration, update or miss-configuration.

Second: Privacy, by disclosing the configuration of the platform.

Third: Scalability, because of the necessary management and changes in the configuration of the TPM during attestation.

Meanwhile, Property-based Attestation (PBA) that is based on various properties which can be described by similar system situations was proposed to overcome the deficiencies of Binary attestation. It also means that two systems, with different configurations, can have similar properties and they can consequently fulfill the same attestation situation [5, 6]. An important issue here is selecting the properties that are always different according to different cases and usages. A property in a platform shows one aspect of behavior of that particular platform with regards to specific requirements. Therefore, various platforms having various components and different configurations can have the same properties which may cause them to fulfill the same requirements. In particular, more flexible approaches to access to the patches and updates of a system can be shown using these properties. There is only one critical issue here, i.e. which properties are more suitable and reasonable in each scenario, and this is definitely dependent upon the usage as well as its case and environment.

The remainder of this paper is organized as follows: A summary of major research previously carried out on TCG attestation and Property-based attestation is given in Section 2. Meanwhile, the design and implementation of the proposed model containing its Property and Protocol are outlined, while how the model works, its implementation and the evaluation are elaborated in Section 3 and 4. Section 5 discusses on the results and discussion, while conclusion is given in Section 6.

2. RELATED WORKS

Binary attestation is a TCG standard model for remote attestation. What we have in the Trusted Platform Module, which is specified in the TCG as binary attestation, is a trusted boot with TPM's cryptographic functionality and secured storage [1, 2]. But as discussed above, Binary Attestation has its own deficiencies [7], and therefore, several researchers,

such as Jonathan Poritz et al. [8] and AhmadReza Sadeghi and Christian Stueble [5], have attempted to explain Property-based approaches with the aim to overcome the lack of flexibility, as well as its privacy and scalability. The general concept behind the Property-based Attestation (PBA) is that instead of attesting hash values of binaries, sample properties showing the behavior of a program or system were attested. Even if the binary data have been changed in each scenario, these properties remain stable or they do not change.

Yan-Li et al. [9] stated some of the advantages of using properties for attestation in trusted platforms, as follows:

(a) Properties do not reveal implementation details of a system and can hide system vulnerabilities.

(b) Properties provide a certain level of privacy by not identifying the components.

(c) During any updates, properties may not change in comparison with hash values.

(d) Properties are easier to understand and can be used to write meaningful access control policies rather than using a series of binary values codes.

Meanwhile, Aarthi Nagarajan et al. [10] conducted a recent researche on property-based attestation that offers a summary of some models in the Property-based area and also shows the characteristics of a property in the PBA. Their contribution has two folds; in the first part, they analyzed different models of property-based attestation mechanisms which have been discussed in the recent years. They also categorized and named these mechanisms as derivation-based, delegation-based and enforcement-based. They analyzed each of these categories by giving a particular focus on their limitations. In the second part, they provided a list of challenges involved in choosing Property for property-based attestation. Based on the categories of attestation model in [10], the model of the current paper is based on Delegation-Based Approaches and the category of Certificate-Based Attestation so these models are described according to the discussion given to them in the literature.

What is important in the certificate-based attestation is that another trusted third party, who is completely trusted by the two parties of communication, has certified the desired properties. The verified machine proves that the trusted third party that is trusted by the verifier will issue the property certificate [5].

Liqun Chen et al. [6] proposed a provably secure Property-based Protocol that implements a delegation-based solution with an offline trusted third party. Their protocol also considered verification and then revocation of the properties and invalid configurations, either from a public list or communication between the verified machine and verifier. Just like the model proposed in this study, this model also uses a certification authority to issue the certificate and sign them to represent the property of the system. The state of the TTP is offline and also they do not use TTP for checking or verifying the revoked certificate. In their next work, they improved their model by introducing a PBA model without a trusted third party [11]. In this model with the help of Ring signatures, they can guarantee the security aspects of their model, list as privacy and scalability. They also formalized their model and generalized existing protocols. But their model cannot directly implemented on current TPMs, because of the lack of some commands in the existing TPMs that they used in their model, like "signed commitment".

At the same time, researchers also considered transmitting secure data in a secured channel that is compatible with the trusted computing features in the same framework of the present study. In order to achieve this goal, the researchers refer to the work of Yacine Gasmi et al. [12] who have described secured and flexible mechanisms to establish and maintain Trusted Channels. Based on their work, the X.509v3 certificates were used in the present study to convey the configuration information during key agreement (TLS handshake). Although the present work theoretically uses the concept of Yacine Gasmi et al. [12], its concrete model is closer to the work of Kenneth Goldman et al. [13] who defined a model to link specific properties of a remote system through TPM-based attestation in order to secure tunnel endpoints and counter attacks where a compromised authenticated SSL endpoint relays onto another system by TPM-based attestation.

The use of certificates to issue the properties is currently being mentioned in some approaches (e.g. Chen, [6] and Yan-Li, [9]). Certificates of the system that have all required signatures and public keys can be one of the indicators of reliability of the system and ca be used to assure the trustworthiness of a party[14, 15]. Certificates that use in this study can be categorized as Identifier properties as it is described in [10]. None of the previous models on PBA has completely been designed as a protocol to use certificates as properties to address property-based attestation, so we need to have a protocol that can assure the trustworthiness of the proposed model based on these certificates as properties.

In this paper, a protocol is proposed with the ability to show the real implementation of property-based attestation (PBA) in a small network consists of one server and one client with the existence of Privacy CA and TTP by using current certificates of the system to generate the property. Considering the deficiencies of the binary attestation, this model does not reveal any configuration of the system so the security issues like privacy, flexibility and scalability can be overcome in this model. The objective of this paper is to propose a property-based attestation model that combines the current certificates of the system (AIK certificate and SSL certificate) to generate a new certificate (Platform Property Certificate) as the property of the system. Meanwhile, the platform property has a link to the Attestation Identity Key (AIK) and the Secure Sockets Layer (SSL) certificates to attest the security of the system. By this method, we use the generated session keys in our protocol in addition with the encryption of the keys to guarantee the integrity of the protocol. Using this newly generated certificate as a property in the model helps the protocol to check the identity of the client by using TPM capabilities and check them by its trusted parties.

3. THE PROPOSED PROPERTY-BASED ATTESTATION MODEL

Considering the current PBA models and their deficiencies, the PBA model in this study is designed based on what are needed in terms of the security and privacy aspects. In this section, the content of the property in this model is first investigated and this will be followed by the design of the proposed property-based attestation model.

3.1 Definition of Property

A property in the PBA model can be attributed to any behavior or characteristic of a specific hardware or software component in a platform. It may also be attributed to the

entire platform as a whole. This makes it more difficult to have any restriction on the scope of what can be defined as a property. Actually, everything in a platform can be defined as a property.

Nagarajan et al. [10] showed the challenges in choosing the property for a PBA model. According to them, some kinds of properties that can be selected in different scenarios are as follows:

- Properties as security services
- Information flow-based properties
- Properties as security functions
- Properties as security policies
- Properties as implementation constraints
- Properties as non-security functions
- Properties as identifiers

Identity of a user can also be a property of the system , some username and passwords and also some system keys that signed by the trusted parties of the system can be considered as properties. In this issue, we can consider certificates of the system as an identifier of the hardware or software of the system and can be considered as a property of the system.

3.2 Using Certificates as Property

The various public key infrastructure issues such as identity, revocation and application specific PKI can be condensed into a set of recommendations for working with certificates [14, 15].

Identity: The first step is defining an identity. In order to define an identity, one should choose a locally meaningful identifier such as a user name, e-mail address, account or employee number, or anything of the kind. Choosing a distinguished name as a meaningful identity is condemned to fail. A locally meaningful identity is not necessarily something which is meaningful to humans. Certificates can work as the local identifier of the system and can keep the credential of the system hidden by not revealing the entire configuration of a system. They have the required signature that proofs the trust-ability of the system and the system can access some common fields of them, e.g. public keys, domains and versions, by extracting their contents.

Revocation: revocation of the certificate is another issue that makes them competitive to choose as a property. By the mechanisms of revocation false certificates, a system can be assured about freshness of the data. The best way to ignore the requirement of certificate revocation is designing the PKI so that it does not require certificate revocation any more. Applying a PKI mechanism that provides the opportunity to certify freshness guarantees is another way to avoid revocation. This shall be considered as a substitution for the necessity of explicit certificate revocation. As an example of this approach a repository that returns only known-good certificates can be mentioned. Providing a direct indication of whether a certificate is valid or not, or a slightly less useful direction that provides a certification revocation list (CRL) response, is the best mechanism of revocation, for instance, the online certificate status protocol can be considered as an example of this approach. CRLs can be applied for cases in which revocation information is of little or no value. Revocation of code-signing certificates can be mentioned as a case that exemplifies this method.

Application-specific PKIs: It is much easier to work with Certificates and PKIs specifically designed to address a particular problem rather than an onesize-(mis)fits-all PKI design. Simple public key infrastructure (SPKI) certificates, as an example of this method, binds a public key to an authorization to perform an action. On the other hand, X.509, binds a key to an often meaningless identity that must be mapped by means of some unspecified means of authorization. In this case if the goal is to authorize a particular action or grant a capability, SPKI is ideal. The situation is also compatible when two or more parties have an established relationship. For instance, by having the user copy the required public keys to where they're needed, an approach feasible for its application domain, the secure shell protocol avoids dependence on a PKI. Even PKI-less public-key encryption shall be considered unnecessary in some cases. In this case a means of distributing and managing certificates which is not also covered in a formal standard, meets the requirements of being utilized.

In the proposed model, the property that is taken into consideration is the combination of the SSL certificate and AIK certificate, as illustrated below: The designed Platform Property Certificate is generated as the newly generated certificate using the current SSL and AIK certificates of each communication based on [13]. Figure 1 shows details of the property that was designed in the proposed model known as the Platform Property Certificate.

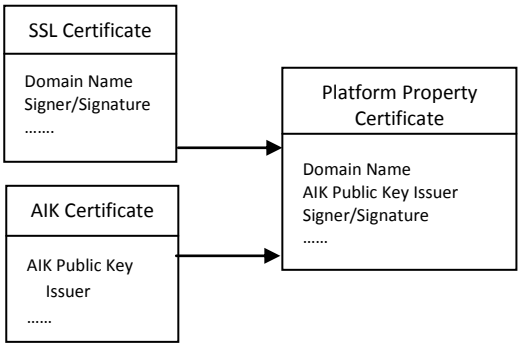

Figure 1: The proposed property of the model

In this study, three characteristics are defined for the proposed property:

- The SSL certificate fetches its domain name in the new certificate which will be proposed to provide the availability of the data in the same domain.
- The AIK certificate that is issued by the Privacy CA fetches its public key issuer to the proposed property.
- The proposed platform certificate should be signed by a privacy CA to consider the authenticity of the proposed property.

3.3 Dataflow and system architecture of the proposed PBA model

This model consists of four parties who need to communicate with each other to implement the attestation function. These parties include the client (i.e. the challenging party of the model), server (which does the attester function of the model), trusted third party (who acts as a trusted component for checking the certificates of the system) and Privacy-CA (that

issues the AIK certificate). As depicted in Figure 2, the general framework of the system can be illustrated as follows:

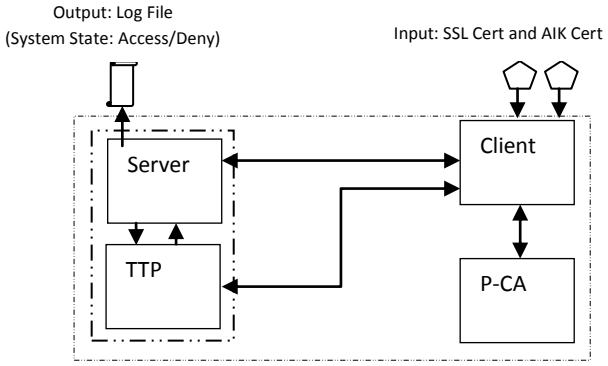

Figure 2: General Framework of the system

As it is shown in Figure 2 because of the advantages of designing the TTP part inside the Server part, the TTP and the server have been taken into consideration to implement in the same machine. Therefore, in the real implementation, there are three machines and each one has a windows platform.

The process starts with sending a 'Hello' message as it has been shown in Figure 3, while an encrypted Nonce from the client to the server in step (1) indicates that the client wants to start the communication. Nonce is a random number that is generated by the random number generator of the TPM. The process of generating a unique nonce can be also done using a simple nonce generation algorithm.

In response to this, the server in step 2 sends its public key with the encrypted server nonce, back to the client. Having the server's public key, the client also has the SSL certificate which is issued by the SSL certifier. The client will then send this certificate (step 3) to the TTP to determine its trustworthiness by checking the revoked certificates list in the TTP. After judging the trustworthiness of the SSL certificate, if it is acceptable the TTP will store this certificate in its trusted SSL certificate list. In the next steps (4 and 5), the AIK certificate is generated. In order to issue AIK certificate, the client sends a public and private key pair that is issued during a certificate generation process, to the Privacy-CA. The rule of the Privacy-CA in this protocol is issuing the AIK certificate. This party can also be a remote party like www.privacyca.com which issues AIK using the EK of the client machine TPM. It is important to note that a dedicated machine that would issue the AIK is used in this work.

The Privacy-CA will then return the AIK to the client and the client will send this AIK certificate (step 6) to the TTP to be stored in the trusted certificate list and to assure its trustworthiness by the TTP. If the TTP cannot accept this certificate, it will store it in the list of revoked certificate and reject it; otherwise, it will resend it to the trusted list of the TTP party.

After sending the AIK certificate to the TTP to checking its trustworthiness, with the AIK and SSL certificates, the client will issue a platform certificate based on the model that has previously been described in section 3.2, to generate a platform property certificate. This certificate is actually a combination of the AIK and SSL certificate; it will get its public key issuer from AIK and obtain its domain name and signature/signer name from the SSL certificate.

In step (7), the client sends the proposed property certificate (marked with Prop* in the figure) to the server. This particular platform property certificate should be checked by the server before it returns the certificate to the TTP.

Meanwhile, the TTP should determine the trustworthiness of this certificate. Having a list of the AIK trusted certificate and a list of the SSL trusted certificate, the TTP will read the platform certificate and check its elements using available certificates that are saved for this transaction as the session keys. If they match, the TTP will then accept the certificate (step 12) and it will be rejected (step 11) if it does not matched.

In the next step, the client sends its nonce and also returns the server nonce that is signed with the server public key to the server (step 9). Then server checks these two nonces, decrypts them with the server private key and matches them with the nonce that it is stored from the first steps of the transaction. If they are the same, the server will accept the nonce (step 10); otherwise, it will reject them (step 13). The verification of nonce is the final step of this attestation to avoid attacks.

Finally, once the server has determined the trustworthiness of the nonce and also has been assured about the trustworthiness of platform certificate, it will allow the client to start the transaction and the attestation result will therefore be "Grant". On the contrary, if TTP does not accept the platform certificate or nonce that is signed with the client is not the same with the one that the server has already stored, the result of the attestation will be "Deny".

The entire process operates within the attestation algorithm; for monitoring purposes, however, there is another party that is involved in the attestation process and is known as monitoring agent who can be installed in the client machine or in the server machine. The responsibility of this party is to store all the transactions of the system. It means that each party will send its status to the monitoring agent when it wants to do any steps of the transactions and the monitoring agent will then show it in its desktop area and also store in a logs file. This can guarantee the integrity of the process of the system because each party should always update its status with the monitoring agent.

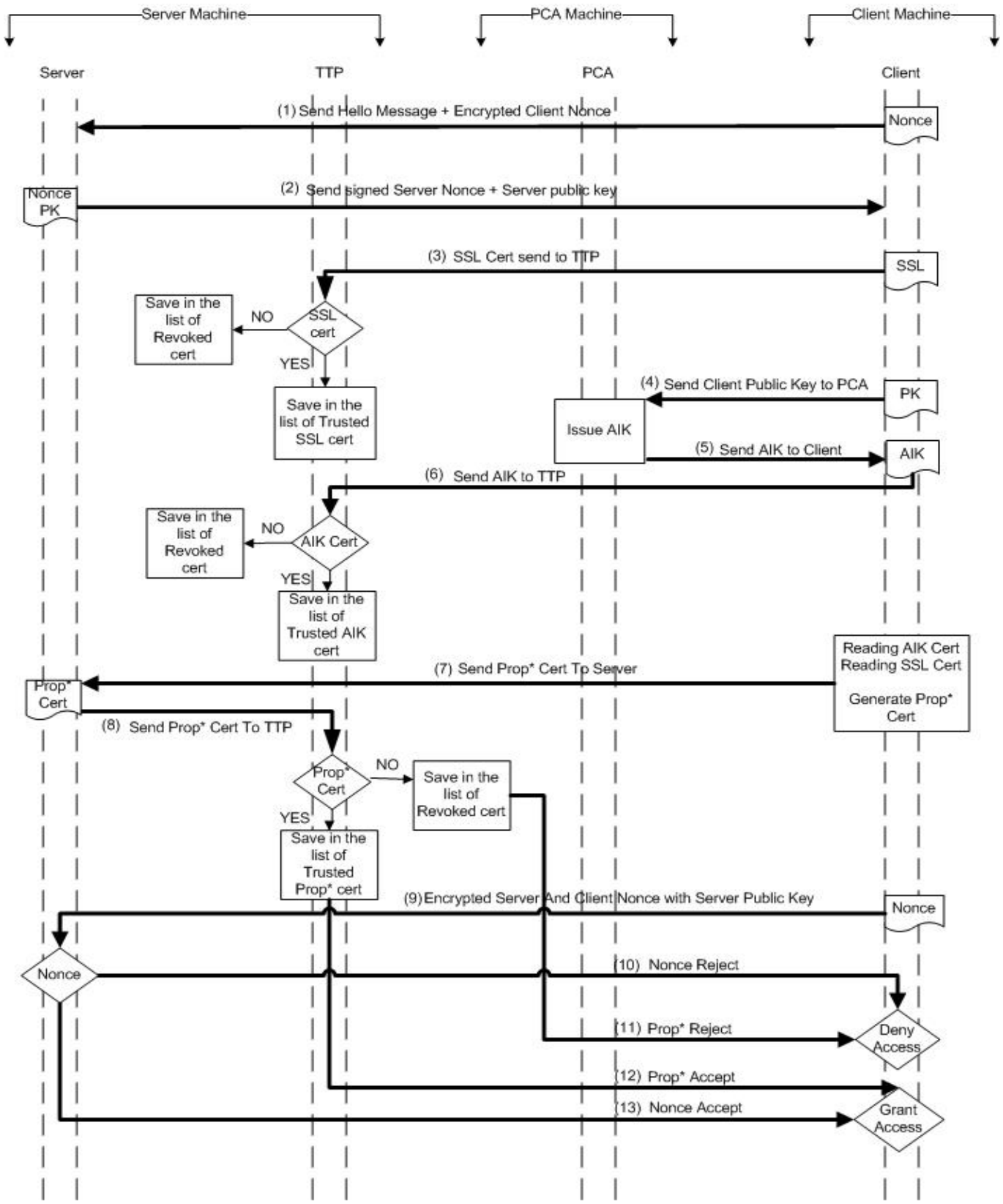

Figure 3: Dataflow of the proposed PBA Protocol

3.4 Key Distribution Protocol

The informal description of the protocol dataflow is coming in Figure 4. Based on [16] for using abstraction to avoid explaining the detail of the local TPM functions, we also stick to the numbering used in [17] to avoid confusion. This model is considered to protect the integrity reporting protocol against masquerading attacks. We enhance it with a key agreement

protocol. Figure 4 shows the integrity reporting protocol with the extension to use Diffie-Hellman parameters.

1	C	S	Signed$_{PK\,C}$ ($Nonce_{Client}$)
2	S	C	Signed$_{PK\,S}$ ($Nonce_{Server}$)
3	C	TTP	$SSL\ Cert_{Session\ S\ and\ C}$
	TTP		Check Certificate(SSL)
4	C	PCA	PK_C+SK_C
	PCA		Generate Key(AIK)
5	PCA	C	AIK, PK_c , SK_c
6	C	TTP	AIK, PK_c
	TTP		Check Certificate(AIK)
	C		Generate key $(Prop^x)_{AIK\,,SSL}$
7	C	S	Signed$_{PK\,S}$ ($Prop^x$, $Nonce_{Client}$, $Nonce_{Server}$)
8	S	TTP	$Prop^x$
	TTP		Check Certificate($Prop^x$)
	S		Auth($Prop^x$, $Nonce_{Client}$, $Nonce_{Server}$, PK_c , PK_s)
9-13	S	C	Access Grant or Deny

Figure 4: Key distribution Protocol

The numbering in the left hand of Figure 4 shows the process that is fully described in Figure 3. As it is mentioned in section 3.3, attestation is started with sending signed nonce from client to server and sending back from server to client (step 1 and 2), sending SSL certificate from client to TTP and then check it by TTP (step 3), generating and sending AIK by Privacy-CA (step 4 and 5), checking AIK by TTP and generate Prop* by client (step 6), sending signed nonces and also prop* from client to server (step 7), checking prop* by the TTP (step 8), and finally determining the validity of nonce and platform property certificate by server in step 9-13.

3.5 A simple Masquerading attack model

In the case of masquerading attack, Figure 5 depicts the attack against the integrity reporting protocol based on [18]. If S (Server) wants to securely validate the integrity of the attesting malicious system A (Attacker), the malicious system itself transfers all messages from S to the honest C (Client). The simple schematic of this attack is illustrated in Figure 5 by reducing it to the transferred messages. Server wants to attest the client before protected data is transferred. Step 5 and 6 shows that platform A and platform C are working collaboratively in the way that platform A is authenticated by the provided information through platform C. Since the protocol is only authenticated a certain user, by using a secure mutual authenticated SSL channel the attack cannot be prevented. Extracting X.509 certificate and keys are possible in mutual authentication. Then, the server authenticates the client based on the certificates and sends an attestation request to the malicious platform, which answers the request in the previously described manner.

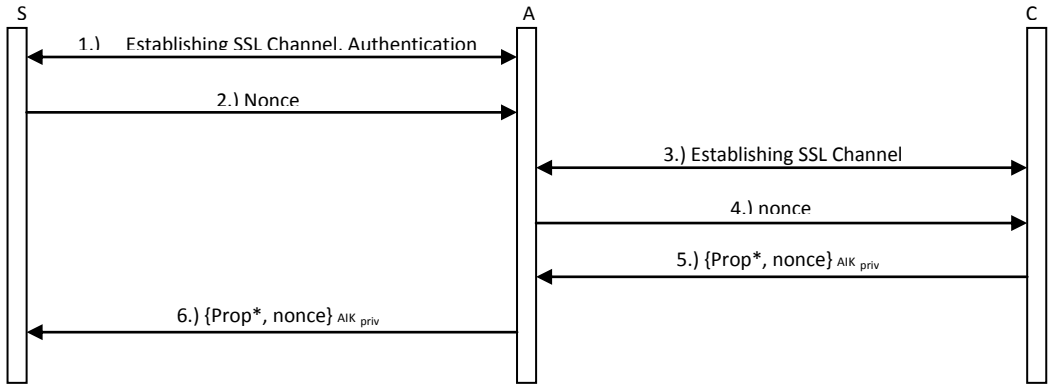

Figure 5: Collaborative masquerading attack on RA based on [18]

4. EVALUATION OF THE MODEL

To reach the objectives of this study, we should prove the validity of the Platform Property and the validity of the proposed Protocol. We also need to show the integrity of our model based on preventing the masquerading attack and the solution for it. Overcome the deficiencies of binary attestation reported as privacy, flexibility and scalability are another goal of this study.

4.1 Measurement criteria of the model

To check the privilege of the model proposed in this study in the term of its flexibility, privacy, scalability and integrity, these parameters should be evaluated in the model. In this section we focus on each of these criteria individually.

Flexibility: is checked by reviewing the result of manipulating PCR of the system.

Privacy: is guaranteed by not revealing the sensitive data.

Scalability: is ensured by not configuring the TPM during the attestation process.

And integrity: is checked by the possibility of affecting masquerading attack to the protocol.

In order to check these criteria, first two different experiments were carried out to study the proposed model.

The first experiment is based on the process of attestation which was done by checking the output log file of the system in the case of valid certificate, whereby the attestation should be turned on, while the process should be properly run.

The second experiment involved running the system using valid and invalid certificates. The valid certificate that comes from the trustable Privacy-CA and SSL certifier must not be

expired or place in the list of revoked certificates, while the certificate that listed in the revocation list of certificates or expired certificates are considered as invalid. Meanwhile, valid certificates will issue a valid property and the result of the system is therefore "Accepted", whereas invalid certificates that lead to invalid property should get the "Deny" response from the system.

4.2 Protocol Discussion and integrity attacks

Since all following messages in the presented protocol are encrypted with the computed session key (SSL key), the protocol prevents an attacker from spoofing his malicious software configuration. In the model described in Figure 3, it is also impossible for A to compute an own session key between him and C since his software is in a compromised state and his TPM is providing malicious platform configurations to C. So access to the private part of the session key that is stored on C is impossible for the malicious platform. All the transferred data in the protocol are encrypted with session keys that are SSL session key or AIK public part that both of them are generated every time in each communication. Based on [15] these session keys cannot be transferred to the malicious host by the platform owner, since the extraction, e.g., by memory dump or by modifying the system software, would lead to a non-conformant system state which will be detected in the attestation phase. In the case of encryption the messages by AIK, attacker cannot decrypt the traffic, as it does not have the private part of the AIK, which is stored in the protected storage of TPM.

5. RESULTS AND DISCUSSION

After running the system and implementing the proposed modules on different machines, using two stated experiments, different results of the system were obtained and these will be discussed in this section.

The First Experiment: In the first experiment, the system was run with a valid certificate to get its log file. At the same time, attestation should turn on and the system should work properly. Therefore, the system was run to retrieve the result from the output file. The output log file is shown in Appendix (Figure 6).

The Second Experiment: In this experiment, the system was run with valid and invalid certificates. In the case of the valid certificate, the system obtained the same result as above experiment and the attestation function was found to have worked properly and turned on (see Appendix-Figure 6). On the contrary, for invalid certificate, the output log file showed that the attestation function did not permit the system to turn the communication on and the result was "Deny" (see Appendix –Figure 7).

Therefore, the results of the first experiment revealed the feasibility of the proposed protocol which resulted in the case of valid input for the model, the system was found to have worked properly and the attestation result that was monitored by the monitoring agent had also been "accepted. Meanwhile, the second experiment showed the validity of the proposed property; in the case of valid input certificates the property created and accepted by the system, while in the case of invalid input certificates that were not accepted by the system, the result of the model was “deny” and the communication was also not allowed to start.

In the model proposed in this study, the result from attestation was not depended on the values of the PCRs. Therefore, it will not affect the attestation results if the value of the PCRs needs to be changed by any application while running the operating system. Thus, it can be stated that the Property-Based Attestation model has overcome the flexibility deficiency of the Binary attestation model.

There are also Privacy and Scalability problems reported for binary attestation:

The privacy of the proposed model is guaranteed by not sending the exact values of PCRs and does not reveal the content of the Measurement List during attestation mechanism in the model.

And scalability of our model depends on not configuring and managing TPM in property-based attestation mechanism during the attestation process, because the attestation process is not depends on any change of the TPM configuration.

6. CONCLUSION AND FUTURE WORKS

In this paper, a client-server protocol that can guarantee the security of the attestation model has been proposed and this has been done by checking the certificates of the model by another Trusted Third Party. This TTP is fully trusted by both parties of the attestation. The Platform Property certificate was issued that is a dependent and linked certificate to the other certificates of the system. If any of these certificates are considered as invalid or expired or placed in revocation list, the attestation protocol will also reject them and the attestation mechanism will deny the concerned party to access the resources.

The results of the experiments designed in this study show that in the case of valid and invalid input certificates for the proposed model, the proposed property can guarantee the trustworthiness of the system. Thus, the flexibility, privacy and scalability of the proposed model can be guaranteed or assured by comparing it with the designed Binary attestation model. Investigating the integrity aspect of the model shows that the system is not vulnerable to integrity attacks because of generating session keys and using encryption, decryption and signed data during attestation protocol.

In contrast with other models of PBA, that consider formal definition of the model without implement it in the real world scenario, and far from considering the security aspects of the system, current model can be implemented and compatible with the current TPMs and also the released TCG standard configurations.

Certainly there are plenty rooms of optimization in this approach. For instance, current platform property certificate is not the only option that can consider as the property of the model. There can be other properties which could guarantee the security of the system in the proposed property-based attestation model; these include some security services like confidentiality or privacy that can be considered as the property of the system. In addition, time or access control can be mentioned as other suggested properties of the model in different scenarios. Meanwhile, proofing the concept of the proposed property and the designed protocol by using formal grammar can be another important topic for further study.

7. REFERENCES

[1] TCG, Trusted Computing Group. http://www.trustedcomputinggroup.org.

[2] Sadeghi, A.-R., Trusted Computing —Special Aspects and Challenges. In: SOFSEM 2008: Theory and Practice of Computer Science, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, pp. 98-117, Vol 4910, 2008.

[3] Pearson, s., Trusted Computing Platforms: TCPA Technology in context. Book - Prentice Hall PTR 2003.

[4] Vivek Haldar, D.C.a.M.F., Semantic Remote Attestation — A Virtual Machine directed approach to Trusted Computing. In: Proceedings of the 3rd conference on Virtual Machine Research And Technology Symposium - Volume 3. USENIX Association, Berkeley, CA, USA, 3-3, 2004.

[5] AhmadReza Sadeghi , C.S.u., Property-based Attestation for Computing Platforms: Caring about properties, not mechanisms. In: Proceedings of the workshop on New security paradigms (NSPW '04). ACM, New York, NY, USA, 67-77, 2004.

[6] Liqun Chen , R.L., Hans Lohr, Markus Rohe, AhmadReza Sadeghi, and Christian Stuble, A Protocol for Property Based Attestation. in: Proceedings of the first ACM workshop on Scalable trusted computing (STC '06). ACM, New York, NY, USA, 7-16, 2006.

[7] Ulrich Kühn, M.S., and Christian Stüble, Realizing Property-Based Attestation and Sealing with Commonly Available Hard- and Software. In: The Second ACM Workshop on Scalable Trusted Computing (STC'07), 2007.

[8] Jonathan Poritz, M.S., Els Van Herreweghen, Michael Waidner IBM Zurich Research Laboratory Z¨urich, Switzerland, Property Attestation—Scalable and Privacy-friendly Security Assessment of Peer Computers. in: IBM Technical Report, 2004.

[9] CUI Yan-Li, Z.X., Credibility Attestation of Property Remote Attestation Method. In: Second International Conference on Future Information Technology and Management Engineering, IEEE, 2009.

[10] Aarthi Nagarajan, V.V., Michael Hitchens, Eimear Gallery, Property-based Attestation and Trusted Computing: Analysis and Challenges. in: Third International Conference on Network and System Security IEEE Computer Society, 2009.

[11] Liqun Chen, H.L., Mark Manulis, Ahmad-Reza Sadeghi, Property-Based Attestation without a Trusted Third Party. in: Information Security Conference (ISC), 2008.

[12] Yacine Gasmi, A.-R.S., Patrick Stewin, Beyond Secure Channels. in: Proceedings of the ACM workshop on Scalable trusted computing, 2007.

[13] Kenneth Goldman, R.P., Reiner Sailer, Linking Remote Attestation to Secure Tunnel Endpoints. in: Technical Report RC23982, IBM, 2006

[14] Gutmann, P., PKI: it's not dead just resting. In: IEEE Computer Society, (vol. 35 no. 8) pp. 41-49, August 2002.

[15] Jiguo Li, X.H., Yi Mu, Willy Susilo, Qianhong Wu, Constructions of certificate-based signature secure against key replacement attacks. In: Journal of Computer Security, Publisher IOS Press, Computer & Communication Sciences, (vol 18, Number 3) pp. 421-449, May 20, 2010.

[16] Ronald Toegl, G.H., Karin Greimel, Adrian Leung, Raphael C-W., Phan and Roderick Bloem, Formal Analysis of a TPM-Based Secrets Distribution and Storage Scheme. In: The 9th International Conference for Young Computer Scientists. IEEE Computer Society, 2008.

[17] Paul E., Sevin c, M.S., and David Basin, Securing the Distribution and Storage of Secrets with Trusted Platform Modules. In: IFIP International Federation for Information Processing, 2007.

[18] Frederic Stumpf, O.T., Patrick R¨oder, Claudia Eckert, A Robust Integrity Reporting Protocol for Remote Attestation. In: Second Workshop on Advances in Trusted Computing,Tokyo, Japan, November 2006

Appendix

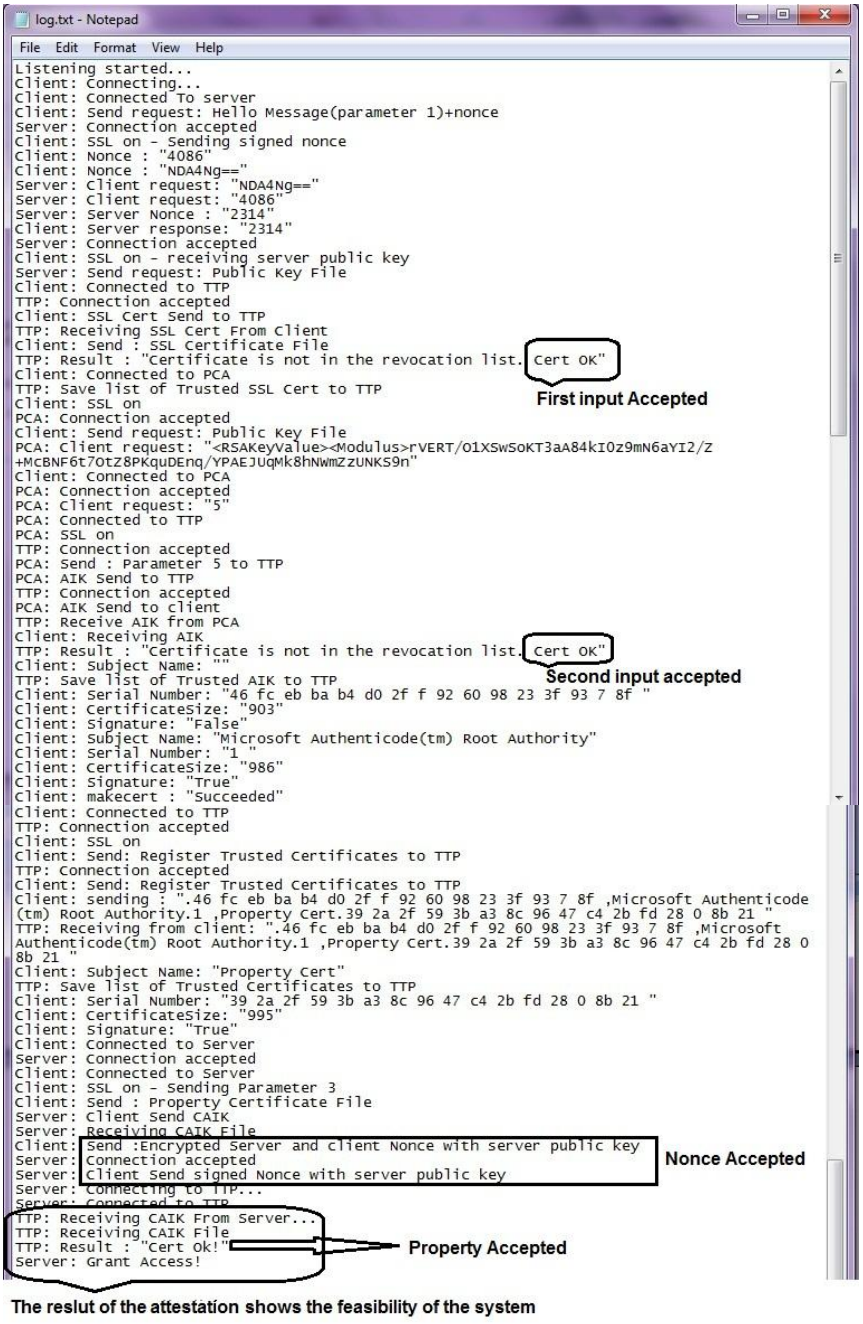

Figure 6 – Results of valid certificate (Grant Access result)

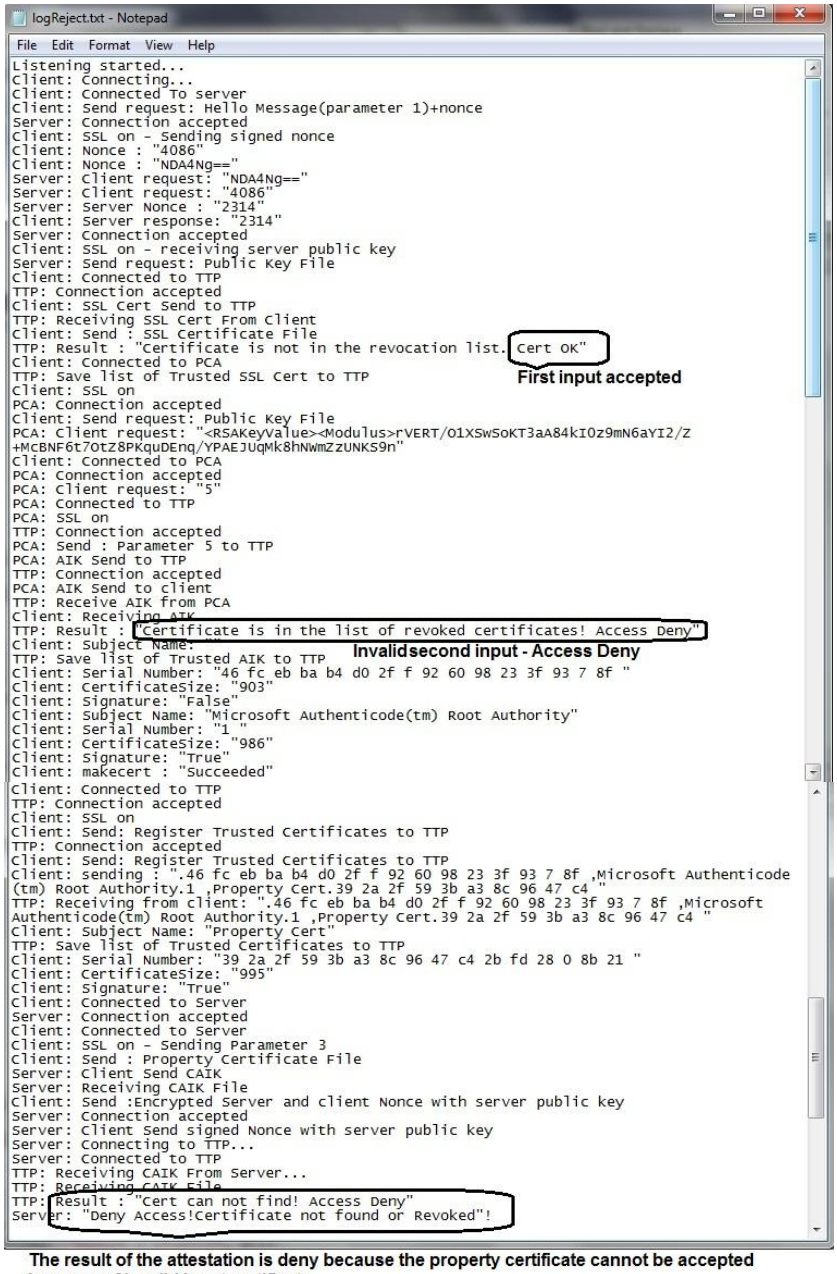

Figure 7 – Result of invalid certificate (Deny Access result)