2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications

A Trusted Remote Attestation Model based on Trusted Computing

Yue Yu, Huaimin Wang, Bo Liu, Gang Yin
National Laboratory for Parallel and Distributed Processing
National University of Defense Technology
Changsha, China
yuyue_whu@foxmail.com, whm_w@163.com, jack_nudt@gmail.com

***Abstract*—Traditional security protocols can not be trusted in some application scenarios of high security level because the endpoints integrity is ignored. In this paper, we propose a novel trusted remote attestation model which combines the secure channel and the integrity measurement architecture of trusted computing. We design and implement a prototype system based on a mature security protocol, Transport Layer Security (TLS) protocol, integrated with integrity report provided by trusted platform module (TPM). The TLS protocol guarantees the security of data exchange process and the integrity report of TPM provides the evidence about the trustworthiness and the security state of the communication endpoints. Compared by traditional approaches, our method is more efficient and can be deployed in large scale systems easily.**

***Keywords*—remote attestation; secure channel; integrity report; trusted computing**

I. INTRODUCTION

Modern era, people are more dependent on the Internet than before, and have increasing demand for service provided by the Internet. Some confidential information requires transmitting in secure channel. The traditional security network protocols, such as Security Socket Layer (SSL) protocol and Transport Layer Security (TLS) protocol, just set up a secure channel in which attackers cannot steal or distort transmitting data[4][5].

However, this kind of secure channel is not integrity because of the exclusion of the endpoints secure state. As a result, if the endpoint is invaded by malicious software, it is possible to appear such embarrassing situation that after terminal identity authentication passing through, establishing connection and secure transmission, data is stolen by malicious codes on the terminal. Take the once wide spread virus, Win32.Huhk.d.7607 called "E-band hiding robber", as an example. The virus can infect the main program of IE browser in the system, IEXPLORER.EXE, after into the system through web Trojan. In this way, when the user using IE browser log in the E-bank and trade, the virus can automatically intercept a lot of related information, including the user's payment card number, password, payee name and other sensitive information. The virus invades user terminal, thus bypassing the secure communication channel established by both sides, which bring the great danger to E-bank users. Therefore, using a secure channel to an end-point of unknown integrity is ultimately futile. In the words of Gene Spafford [1], "using encryption on the Internet is the equivalent of arranging an armored car to deliver credit card information from someone living in a cardboard box to someone living on a park bench."

People have realized that in the face of existing security risks and threats, we not only need a top-down security system design, but also need to bottom up ensure the credibility of computing systems from the terminal. Thus the technology of trusted computing (TC) developed rapidly and has also become a hotspot in academia [2][3]. Reporting integrity information to a remote platform is one of the main goals of TC as proposed by the TCG. There is one security chip named Trusted Platform Module (TPM) integrated into mother board of computing platform [6]. Before every component taking control of main CPU, such as BIOS, MBR, OS Kernel, Application and so on, its characteristic code and configure data must be measured, and the measured value is stored into TPM Platform Configuration Registers (PCR). When the computing platform wants to access some resources in remote entity, remote entity can ask the computing platform to give a security status report. Report data includes computing platform's TPM-based identity information and PCR value. Remote entity evaluates the report data and makes decision to allow computing platform to access the resource or not, which is illustrated in Figure 1.

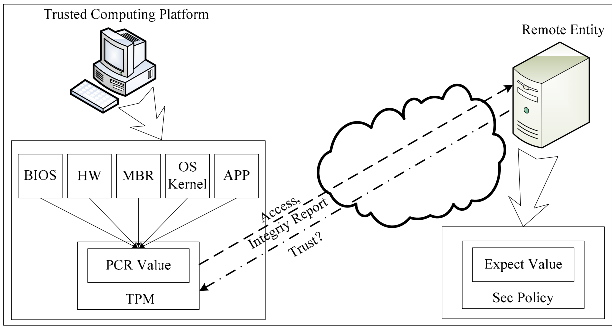


Figure 1. Trusted Computing Platform integrity report

This mechanism has some serious limitations. It has many shortcomings such as inconvenient software upgrading, not adapting to dynamic changes of the system configuration, easy to bind to a special products, and easy to leak platform

978-0-7695-5022-0/13 $26.00 © 2013 IEEE
DOI 10.1109/TrustCom.2013.183

configuration [7]. Moreover, network security protocols which support the TCG remote report are less and combine is not close enough.

The main contribution of this paper is to illustrate how to combine the security channel with integrity measurement architecture of trusted computing. A trusted remote attestation model is proposed and the implementation is given in detail including: (1) TPM integrity report mechanism and related functions; (2) The steps to obtain Platform Identity Key (PIK) of TPM; (3) How to make the integrity policy and configuration of platform; (4) The approach of TLS Handshake Protocol extensions.

The article structure is as follow: Section II analysis some related work; Section III introduces the overall structure and basic framework of the model proposed in this paper; Section IV presents specific implementation of each component of the model. We summarize this paper and make the future research plan in Section V.

II. RELATE WORK

A. Research of TNC

With the development of Internet, trusted computing should not only guarantee the trustworthiness of terminal computing environment, but also extend to network, making it become a trusted computing environment [8]. In May 2004, TCG founded the Trusted Network Connection Sub Group [9] (TNC-SG) which mainly takes charge of the research and setting of Trusted Network Connection (TNC) framework as well as some relevant Standards. Construction of trusted network is extremely tough owing to the complexity of network itself, so TCG firstly takes the relatively elementary problem into account. TNC is the expansion of TPM as well as the combination between mechanism of trusted computing and that of network access control. The TNC architecture [10][11] with trusted computing platform is shown in Figure 2.

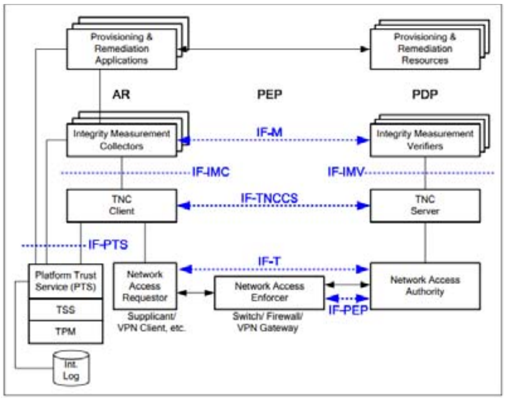

Provisioning & Remediation Applications
Provisioning & Remediation Resources
AR
PEP
PDP
Integrity Measurement Collectors
IF-M
Integrity Measurement Verifiers
IF-IMC
IF-IMV
TNC Client
IF-TNCCS
TNC Server
IF-PTS
IF-T
Platform Trust Service (PTS)
TSS
TPM
Network Access Requestor
Supplicant/ VPN Client, etc.
Network Access Enforcer
Switch/ Firewall/ VPN Gateway
IF-PEP
Network Access Authority
Int. Log

Figure 2. TNC architecture with TPM

Its main idea is as follow. The identity of users is verified before terminal accessing network. If the information of users identify gets through, the identify of terminal platform should be verified. Then, TNC will measure the terminal platform trusted states and if the result of measurement meet the secure policies, it allows terminal to access network, otherwise it will connect the terminal to appointed isolation area and make relevant mending or upgrade. TNC is a method to realize the network access control and that to implement initiative protection which can restrain most of the potential attack before their outburst.

However, there are three relatively pivotal problems of the remote attestation under current TNC architecture which this paper focuses on:

1) One-way trustworthiness evaluation. Before the terminal accessing network, besides providing its own trusted evidence of platform, it should also assess the network, otherwise it can not guarantee credibility of the service acquired from network.

2) Relatively independent of current security system. To realize the access control, TNC devised a set of independent mechanism. As shown in Figure 2, the TNC framework is divided into three entities, three levels and several interface components. The realization of every part is relatively complex and lacks the support of secure protocol. For example there require a mass of information exchange between TNCC and TNCS or TNCC and IMC, however the TNC framework itself has not provided relevant secure protocols.

3) Lacking the good combination with the secure channel. Although TNC framework considers the process that platform reports its integrality to remote entity, it does not combine with existing secure channel to guarantee confidentiality or integrality of platform and etc.

B. Research of TPM-based Remote Attestation

There are many different previous works in the area of remote attestation or access control based on TPM or TC. In [12], a Property-Based Attestation (PBA) mechanism is proposed, which extends the architecture of TC remote report model and includes the property values of the remote side in an attestation. A Trusted Third Party (TTP) translates the actual system configuration into a set of properties and issues certificates for those properties. During the attestation process a remote challenger can decide whether or not the platform security properties meet the requirements of the respective use case. Similarly, Semantic Remote Attestation (SRA) [13] uses language-based techniques to attest high level properties of an application. The proposal is based on the Java Virtual Machine (JVM) environment which is attested by binary attestation itself. Paper [14] presents a Peer-to-Peer access control architecture using TC technology. A trusted reference monitor (TRM) is introduced beyond the trusted hardware. This architecture can enforce an object owner's policy in a client platform by attesting the authenticity of the platform and the integrity and possible properties of the requesting application.

Another kind of related works [15] aims at linking end-point configuration information to secure channels. It combines the TLS and IPSec protocols with the platform integrity measurement and reporting features of TCG Trusted Computing functionalities. However, none of the solutions so far addresses the problem fully [19]. Some are specified in insufficient detail, e.g., [16] do not explain how they exactly establish the linkage to TLS. Others have deficiencies regarding security or trust assumptions that we do not achieve. e.g., [17] aims to prevent "relay attacks" or "MitM attacks", but as shown in [18], these attacks seem still possible. The approach given by [18] overcomes most of the shortcomings identified in related work, but it required costly acquisitions of specific cryptographic hardware and some features of the protocol do not conform to the TLS specification [5], e.g., sending attestation data within the key exchange messages, or including integrity data in session key computation. Changing central message formats or computations of the TLS protocol would result in that the TLS specification must be redefined and backward the compatibility of TLS. The solution in [19] conform the TLS specification better than other works and elaborate the trusted channel in detail.

III. Trusted Remote Attestation Model

Carrying out the trusted remote attestation model which the paper refers to is also based on the PKI, and the model is shown as Figure 3. This model consists of three parts: Privacy Certificate Authority (PCA) Server, Internet Service Requestor (ISR) and Internet Service Provider (ISP).

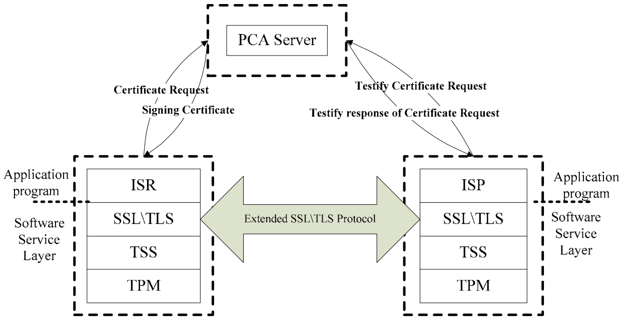


Figure 3. Remote attestation model based on TPM

PCA Server is used to authorize the Platform Identity Key (PIK) to ISQ and ISS and authenticate the mutual identity. The content on the principle of PKI, the reason why we use the PIK, and the way how to apply the PIK will be details in the next Section. In the Figure 3, the application and authentication of PCA Server are both two-way. ISR and ISP use the mutual authentication, and they are both related to certificate request and certificate validation. The Figure only shows the single authentication for convenience.

ISR with TPM guarantees the integrity in the hardware layer and provides the serve of integrity measurement and storage. By using Trusted Software Stack (TSS), ISR provides the interface to call the relevant function of TPM. And then according to extended TLS protocol, build the integrity channel to provide the identification and the report of platform integrity for ISP and testify the ISP platform identity and its platform integrity. The paper calls this layer as Software Service Layer. When the platform integrity authentication succeeds, ISR uses ISR Client Application to request Internet serve to ISP, such as FTP, HTTP and so on.

The process of ISP is similar to ISR, and the difference between them is that ISP provides the Internet source and serves in the application layer by ISP Server Application.

Obviously, compared with the distributed network model this model changes little in the network layer and has no additional system component. It has a better usability and compatibility. The integrity report mechanism of TPM will be introduced in the part A of Section IV. Part B will detail the process of PIK and the way how to make the platform integrity authentication policy will be referred to in part C. Part D will show the way that how ISP builds integrity channel with ISR and it will also details the way to extend the TLS protocol.

IV. Implementation

A. Integrity Report Mechanism of TPM

There is a group of PCRs (Platform Configuration Register) in the TPM [6]. One PCR is only relevant with one type of system special event of computing platform. The integrity value of a trusted computing platform component is stored in PCR. The length of PCR is related to the Hash algorithm used by trusted computing platform. Its length is as long as the length of the calculation of Hash. It needs to compute the integrity value of the component again, when the integrity of this component changes. The computing expression is shown as follow: $\mathsf{PCR_n} = H(\mathsf{PCR_n(Old)} \parallel \mathsf{PCR_n(New)})$

$\mathsf{PCR_n}$ expresses as a new Hash value of nth PCR, $\mathsf{PCR_n(Old)}$ is the original value of nth PCR before measuring, and $\mathsf{PCR_n(New)}$ is the value of the component measured this time. Obviously, the integrity value stored in PCR is able to reflect the change of components integrity and not just reflect one state of the component. If the measured value of several components is stored in the same PCR, it needs to split the integrity measured value of the component joint the value stored in PCR, do the Hash computing and restore the result got in the PCR one by one in the special sequence. When the integrity measured value of the last component is stored, the value in the PCR is the integrity measured value of this component.

Trusted Computing Platform records the information of measurement in the Stored Measurement Log (SML). From the start of platform powering up, the trusted chain $\mathsf{RTM} \rightarrow \mathsf{BIOS} \rightarrow \mathsf{OS\ Loader} \rightarrow \mathsf{OS} \rightarrow \mathsf{Application}$ is recorded.

In the process of remote attestation, the integrity report provided by platform is the SML and its corresponding PCR value. It includes the static configuration of terminal, such as system version, hardware condition. It also includes the dynamic configuration of terminal, such as process condition, network flow information and so on. In that way, the two sides of communication can use this integrity report to verify the static and dynamic integrity of each terminal.

B. PIK certificate for remote attestation

One of the easiest ways to identify a TPM is to verify the serial number which is one-to-one relationship between a TPM. However, the problems of user's privacy should be considered. For example, if a TPM user always accesses some shopping website, his hobbies can be analyzed by his shopping history records.

PKI-based TCM key architecture has been give out in paper [20]. The manufacture generates an endorsement key (EK) during manufacturing stage of TCM. EK is a pair of asymmetric key which is stored in the non-volatile protected storage area in the TCM. One TCM can own only one EK during its life cycle. In order to protect privacy of users, the platform doesn't directly use EK to sign for identity authentication. Users have to request the TCM to generate a PIK for every special Web application, and apply for PIK certificate from PCA based on EK. The privacy of the corresponding relationship between EK and PIK is protected by PCA. A TCM and its host platform can have multiple PIKs. Thus, users can use different PIK for identity authentication in different situations in order to achieve the purpose of the platform identity privacy protection.

In this paper, the handshake protocol of TLS needs to be expanded which PIK certificate should be used. The process of applying a traditional X509.v3 certificate must change to bind with TPM. The generating of PIK, applying for, procedure of signature and activation PIK certificate are shown as Figure 4.

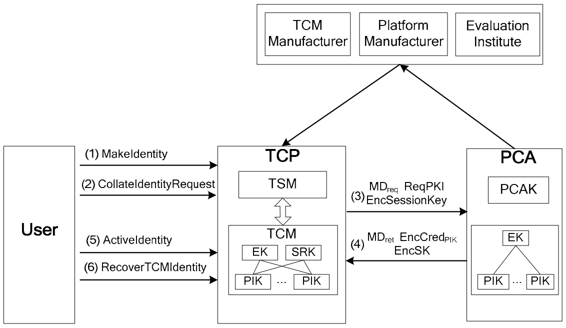


Figure 4. Procedures of generating of PIK, applying for, signing and activating a PIK certificate

1) Trusted computing platform (TCP) User sends a command `MakeIdentity` to Trusted Cryptography Module (TCM) via TCM Service Module (TSM). The TCM generate a PIK and encrypt the private key of PIK with SRK for protection. Then the TCM use the private key of PIK to sign the public key of private CA and the message digest value of the public key of PIK, the signature is $\mathsf{PIKSign} = Sign(\mathsf{PIK}_{\mathsf{Pri}}, H(\mathsf{PCAK}_{\mathsf{Pub}} \parallel \mathsf{PIK}_{\mathsf{Pub}}))$. The TCM return the public key $\mathsf{PIK}_{\mathsf{Pub}}$ and $\mathsf{PIKSign}$.

2) TCP User sends a request `CollateIdentityRequest` to TSM for getting EK certificate $\mathsf{Cred}_{\mathsf{EK}}$ and property certificates $\mathsf{PtyCred}_{\mathsf{TCM}}$ and $\mathsf{PtyCred}_{\mathsf{TCP}}$ which are signed to TCM and its host platform by evaluation institute and manufacturer.

3) TCP User uses the TPM as source for random values to generate a symmetric $\mathsf{SessionKey}$, because the Random Number Generator (RNG) is considered as true random generator. Meanwhile, User makes out his information Info for a traditional X509.v3 digital certificate request. Then, ReqPKI can be achieved by using symmetric encryption algorithm, e.g. 3DES, and $\mathsf{SessionKey}$. The formula of ReqPKI is:

$$\mathsf{ReqPKI} = SEnc(\mathsf{SessionKey}, \mathsf{PIK}_{\mathsf{pub}} \parallel \mathsf{PIKSign} \parallel \mathsf{Cred}_{\mathsf{EK}} \\ \parallel \mathsf{PtyCred}_{\mathsf{TCM}} \parallel \mathsf{PtyCred}_{\mathsf{TCP}} \parallel \mathsf{Info})$$

TCP User uses asymmetric encryption algorithm to encrypt $\mathsf{SessionKey}$ by the public key of PCA, $\mathsf{EncSessionKey} = AEnc(\mathsf{PCAK}_{\mathsf{Pub}}, \mathsf{SessionKey})$. $\mathsf{MD}_{\mathsf{req}}$ is the message digest of $\mathsf{EncSessionKey}$ and ReqPKI , $\mathsf{MD}_{\mathsf{req}} = H(\mathsf{EncSessionKey} \parallel \mathsf{ReqPKI})$.

Finally, $\mathsf{MD}_{\mathsf{req}}$, $\mathsf{EncSessionKey}$ and ReqPKI are sent to PCA to apply PIK certificate.

4) PCA verifies data integrity by $\mathsf{MD}_{\mathsf{req}}$ and use the private key $\mathsf{PCAK}_{\mathsf{pri}}$ to decrypt $\mathsf{EncSessionKey}$ to get $\mathsf{SessionKey}$, $\mathsf{SessionKey} = ADec(\mathsf{PCAK}_{\mathsf{pri}}, \mathsf{SessionKey})$.

ReqPKI can be decrypted as the follow formula:

$$\mathsf{PIK}_{\mathsf{pub}} \parallel \mathsf{PIKSign} \parallel \mathsf{Cred}_{\mathsf{EK}} \parallel \mathsf{PtyCred}_{\mathsf{TCM}} \parallel \mathsf{PtyCred}_{\mathsf{TCP}} \\ = SDec(\mathsf{SessionKey}, \mathsf{ReqPIK})\ .$$

Furthermore, PCA verifies $\mathsf{PtyCred}_{\mathsf{TCP}}$, $\mathsf{PtyCred}_{\mathsf{TCM}}$ and $\mathsf{Cred}_{\mathsf{EK}}$, then $\mathsf{PIKSign}$ will be verified by PCA using $\mathsf{PIK}_{\mathsf{Pub}}$. After these verifications, PCA signs PIK certificate, then generates randomly a symmetric key SK and use it to encrypt $\mathsf{Cred}_{\mathsf{PIK}}$ to get $\mathsf{EncCred}_{\mathsf{PIK}} = SEnc(\mathsf{SK}, \mathsf{Cred}_{\mathsf{PIK}})$. At last, it use the public key of EK to encrypt SK to get the $\mathsf{EncSK} = AEnc(\mathsf{EK}_{\mathsf{Pub}}, \mathsf{SK})$ and compute hash value of EncSK and $\mathsf{EncCred}_{\mathsf{PIK}}$, $\mathsf{MD}_{\mathsf{ret}} = H(\mathsf{EncSK} \parallel \mathsf{EncCred}_{\mathsf{PIK}})$. PCA sends the data of $\mathsf{MD}_{\mathsf{ret}}$, $\mathsf{EncCred}_{\mathsf{PIK}}$ and EncSK to TCP.

5) TCP User verifies data integrity by $\mathsf{MD}_{\mathsf{ret}}$ and sends a command `ActiveIdentity` to TCM via TSM. TCM can use the private key of EK to decrypt EncSK, $\mathsf{SK} = ADec(\mathsf{EK}_{\mathsf{pri}}, \mathsf{EncSK})$, while only the valid TCP can get the correct SK so we can confirm the relationship of PIK and TCP User. TCM returns SK.

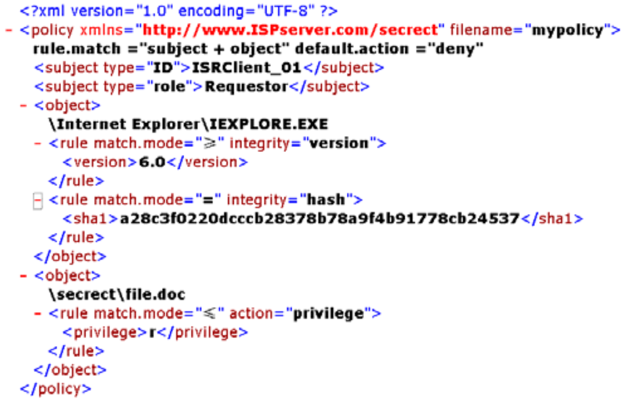
(6) TCP User sends a request `RecoverTCMIdentity` to TSM, TSM uses SK, returned by TCM, to decrypt $\mathsf{EncCred}_{\mathsf{PIK}}$ and gets PIK certificate $\mathsf{Cred}_{\mathsf{PIK}} = SDec(\mathsf{SK}, \mathsf{EncCred}_{\mathsf{PIK}})$.

From what has been discussed above, a TCP User can get a PIK certificate. In this paper, ISR and ISP can apply different kinds of PIKs certificates for various specific situations.

C. Platform Integrity Policy Specification

If ISR and ISP, in Figure 3, want to verify whether the integrality of each other can accord with the their own requirement, firstly they need to define their own configuration requirement of integrity, that is platform integrity policy, afterwards they ask for and verify relevant integrity reports each other. Owing to the possibility that both sides of communication may use inconsistent operating system, the manifestation of integrity policy must consider the otherness and compatibility of heterogeneous computing environment which seems more important especially under the environment of distributed computation. The Figure 5 shows an example of platform integrity policy based on XML.

In Figure 5, subject and object match strictly and default action ="deny" indicates under the state of default any request will be refused except for the below regulation. Considering the fact that remote attestation is bidirectional, subject may be ISR or ISP and the configuration and match method are surely different in those two situations. So subject is identified by "ID", "role" is a changeable element which use to indicate web service requester or web service server. If the platform is Windows System, the first subject is IE. There are two integrity rules about IE. The first one is version which must be above 6.0. This rule can guarantee the stability for IE. Because generally speaking, higher version software is accompanied by fewer bugs and it can run more steadily. The second rule is the hash value of IE. We can choose different hash algorithm according to the different strength here. IE has different hash value for different version, a simple is shown in Figure 5 which is a hash value for IE6.0 in experiment computer computed by SHA-1. This rule ensures data integrity for IE. Because when malicious code infects IE, the hash value will change. The second object is a file which is in the folder “secret”. Its integrity rule is an action configuration which define that the access privilege of this file is read only.



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <policy xmlns="http://www.ISPserver.com/secrect" filename="mypolicy">
   rule.match ="subject + object" default.action ="deny"
   <subject type="ID">ISRClient_01</subject>
   <subject type="role">Requestor</subject>
 - <object>
     \Internet Explorer\IEXPLORE.EXE
   - <rule match.mode="≥" integrity="version">
       <version>6.0</version>
     </rule>
   - <rule match.mode="=" integrity="hash">
       <sha1>a28c3f0220dcccb28378b78a9f4b91778cb24537</sha1>
     </rule>
   </object>
 - <object>
     \secrect\file.doc
   - <rule match.mode="≤" action="privilege">
       <privilege>r</privilege>
     </rule>
   </object>
  </policy>
```

Figure 5. Example for platform integrity policy

According to the description above, the representation of the integrity policy in this paper is flexible. Users can extend their own integrity policies according to different web applications and access control requirements.

D. Handshake Protocol extensions of TLS

The fundamental method of remote attestation is to expend trusted chain to network and one outstanding implementation of the model is provided in paper [19]. The protocol of remote attestation in this paper is based on it. With the Handshake Protocol extensions of TLS, besides the execution of the traditional function, the terminal integrity verification is involved, as shown in the Figure 6.

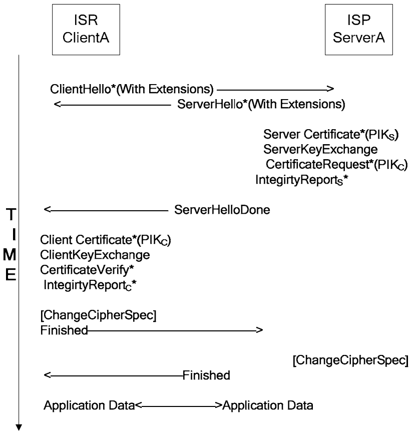


Figure 6. The process of Handshake Protocol extensions of TLS

The first step, the difference between the Hello message, the ISR Client sends to the ISP Server, and the traditional one is that random number $\mathsf{nonce}_{\mathsf{cTPM}}$ of $\mathsf{ClientHello}$ created by TPM Random number generator is to boycott replay attack. The Attestation Extension Demand $\mathsf{AttestExtD_c}$ of the client is contained in $\mathsf{ClientHello(With\ Extensions)}$. $\mathsf{AttestExtD_c}$ is used to transfer the configuration strategy of the client’s platform, and rule the integrity of the other platforms should have, the structure has been stated in Section C.

The expression of $\mathsf{ClientHello(With\ Extensions)}$ is as follow: $\mathsf{ClientHello(With\ Extensions)} = \mathsf{ClientHello} + \mathsf{nonce}_{\mathsf{cTPM}} + \mathsf{AttestExtD_c}$

The second step, Internet Service Provider (ISP) Server sends $\mathsf{AttestExtD_s}$ to Internet Service Requestor (ISR) Client and informs ISR Client the evidence of integrity needed to be provided. At the same time, ISP Server starts to adjust the configuration of the platform dynamically according to the $\mathsf{AttestExtD_c}$ received from ISR Client. In fact, the server should have the capability to work for several clients, but requirements of clients may cause conflicts. For example, Client_01 just allow the server to open the Port 8080, however Client_02 may want to get service from the server via the Port 3389. In this case, the server virtualizes the resources and environment that can meet the client requirements and provide the corresponding services by taking advantage of virtualization technology.

The expression of $\mathsf{ServerHello(With\ Extensions)}$ is as follow: $\mathsf{ServerHello(With\ Extensions)} = \mathsf{ServerHello} + \mathsf{nonce}_{\mathsf{sTPM}} + \mathsf{AttestExtD_s}$

The third step, ISP Server sends its digital certificate (DC), $\mathsf{ServerKeyExchange}$ and $\mathsf{CertificateRequest}$ messages to ISR Client, which is similar to the traditional TLS protocol. However, $\mathsf{CertificateRequest}$ message is necessary in this step which is different from traditional TLS protocol. The digital certificate (DC) is the PIK certificate which is applied as the step described in Section B. Then ISP Server sends platform integrity report $\mathsf{IntegrityReport_s}$ to ISR Client to prove that it meets requirements. The structure of $\mathsf{IntegrityReport_s}$ is: $\mathsf{IntegrityReport_s} = \mathsf{PCR_{sn}} + \mathsf{SML_{sn}} + Sign(\mathsf{PIK_s}, H(\mathsf{PCR_{sn}} \parallel \mathsf{SML_{sn}}))$

$\mathsf{PCR_{sn}}$ and $\mathsf{SML_{sn}}$ are matched by the measurement of TPM for which the virtual environment serve. ISP Server uses the private key of PIK to sign the Hash value of $\mathsf{PCR_{sn}}$ and $\mathsf{SML_{sn}}$. After the transmission of $\mathsf{IntegrityReport_{sr}}$ to ISR Client, message $\mathsf{ServerHelloDone}$ would be sent as the end signal of this stage.

The forth step, which is similar to the traditional TLS protocol. Here we mainly point the differences. ISR Client firstly checks the validity of the PIK Certificate of ISP Server, and then checks the authenticity and integrity of $\mathsf{IntegrityReport_s}$ using the public key of PIK certificate to make sure the data comes from ISP Server instead of being modified by attacker. After that, check the validity of the congruent relationship between $\mathsf{PCR_{sn}}$ and $\mathsf{SML_{sn}}$. At last, decide whether the contents of $\mathsf{SML_{sn}}$ are accorded with integrity policy defined by itself. If an error or authentication failure comes up during the above steps, a fatal error messages would be returned to opposite side and the connection will be cut off. After all the authentications get through, ISR Client sends its $\mathsf{IntegrityReport_c}$ to ISP Server and then the message $\mathsf{Finished}$ as the end. The structure of $\mathsf{IntegrityReport_c}$ is : $\mathsf{IntegrityReport_c} = \mathsf{PCR_{cn}} + \mathsf{SML_{cn}} + Sign(\mathsf{PIK_c}, H(\mathsf{PCR_{cn}} \parallel \mathsf{SML_{cn}}))$

After that, the identity authentication and the integrity authentication of platform of ISR Client are similar to the forth step, and the protocol it used accords to the traditional TLS protocol. Here will not give unnecessary details. Finally, there is some need to point that the transportation of the integrity reports are in plaintext, and what's more, both sides have made autograph using their private key of PIK. When attackers intercept and capture these messages, it is possible to dig out the private information of the platform. Such as which kind of web service ISR Client with PIK_{web} often requires and how is the configuration of platform integrity and so on. To sum up, the platform should ask for different PIK certificates according to different applications, and update the certificate in time to make sure the lifetime of each PIK certificate is reasonable.

V. CONCLUSION AND FUTURE WORK

In this paper, a trusted remote attestation model has been proposed. The model combines the secure channel and the integrity measurement architecture of trusted computing. The secure channel guarantees the security of data exchange process and the integrity report of Trusted Platform Module provides the evidence about the trustworthiness of the communication endpoints. The implementation is described in detail including: (1) TPM integrity report mechanism and related functions; (2) The steps to obtain PIK certificate of TPM; (3) How to make the integrity policy and configuration of platform; (4) The approach of TLS Handshake Protocol extensions. The approach according with the trusted computing technology conforms to the specification of TLS. Thus, the paper assumed that a TPM is only vulnerable to hardware attacks.

In a next step, we plan to give a formalization security analysis of our method and test the experiment system by prevalent malicious codes in the Internet. Then, we consider adjusting the model to the requirements of distributed computing and clouding computing. Moreover, with the development of Trusted Computing technology, the model should be extended by runtime integrity measurement and dynamic attestation agents. In addition, how to combine Trusted Computing technology and other security protocols, e.g. SSH and IPSec, is a challenge research work as well. Last but not least, we plan to demonstrate the effectiveness of our method with quantitative evaluation.

ACKNOWLEDGMENT

This research is supported by the National High Technology Research and Development Program of China (Grant No. 2012AA011201) and the Postgraduate Innovation Fund of University of Defense Technology (Grant No.S120602). Many thanks to Xiang Li, Tao Wang, Tianzuo Wang, and Zhefeng Xiao for discussion and their suggestion and works on data collecting.

REFERENCES

[1] G. Spafford, *Attributed to in Risks Digest 19.37 review of @LARGE*, by David H. Freedman and Charles C. Mann, Sept. 1997.

[2] Sheng C X, Zhang H G, Wang H M, et al, "Research and development of trusted computing," *Science China: Information Science*, vol. 40, pp. 139-166, 2010. (In Chinese)

[3] Shen C X, Zhang H G, Feng D G, et al. Survey of information security [J]. *Science China: Information Science*, vol. 50, pp. 273–298, 2007.

[4] T. Dierks and E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.1*. IETF RFC 4346, Apr. 2006.

[5] S. B.-W. et al, *Transport Layer Security (TLS) Extensions*, IETF RFC 3546, June. 2003.

[6] Trusted Computing Group. *TCG specification architecture overview, specification revision 1.4* [EB/OL], http://www.trustedcomputinggroup.org/files/resource_files/AC652DE1-1D09-3519-ADA026A0C05CFAC2/TCG_1_4_Architecture_Overview.pdf, 2010-06-12.

[7] E Shi, A Perrig, L van Doorn. Bind, "A fine-grained attestation service for secure distributed systems," in *Proc. the IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 2005.

[8] Zhang H G, Chen L, Zhang L Q, "Research on Trusted Network Connection," *Chinese Journal of Computers*, vol. 33, pp. 706-717, Mar. 2010. (In Chinese)

[9] TNC Web Site. [Online]. Available: https://www.trustedcomputinggroup.org/network/

[10] TCG Specification Trusted Network Connect -TNC Architecture for Interoperability Revision 1.1 [EB/OL], Trusted Computing Group, May. 2006.

[11] TCG Trusted Network Connect TNC Architecture for Interoperability Specification Version 1.4 [EB/OL], May. 2009. http://www.trustedcomputinggroup.org/resources/tcg_architecture_overview_version_14.pdf.

[12] A. Sadeghi and C. Stuble, Property-based attestation for computing platforms: Caring about properties, not mechanisms, in *Proc. NSPW*, 2004.

[13] V. Haldar, D. Chandra, and M. Franz, Semantic remote attestation - a virtual machine directed approach to trusted computing, in *Proc. USENIX*, 2004.

[14] Sandhu R, Xinwen Z, Peer-to-Peer Access Control Architecture Using Trusted Computing Technology, in *Proc. SACMAT'05*, 2005.

[15] Goldman K, Perez R, Sailer R, Linking remote attestation to secure tunnel endpoints, in *Proc. STC '06 Proceedings of the first ACM workshop on Scalable trusted computing*, Nov. 2006.

[16] Sadeghi A-R, Stuble C, Wolf M, et al. Enabling Fairer Digital Rights Management with Trusted Computing, in *Proc. Information Security Conference*, 2007.

[17] Stumpf F, Tafreschi O, Roder P, et al, A robust Integrity Reporting Protocol for Remote Attestation, in *Proc. WATC '06*, Dec. 2006.

[18] Gasmi Y, Sadeghi A-R, Stewin P, et al, Beyond Secure Channels, in *Proc. STC '07*, 2007.

[19] Armknecht F, Gasmi Y, Sadeghi A-R, et al, An Efficient Implementation of Trusted Channels based on OpenSSL, *in Proc. STC '08*, Oct. 2008.

[20] Fajiang Yu, Tong Li, Yang Lin, et al, Hierarchical-CPK-Based Trusted Computing Cryptography Scheme, in *Proc. ATC 2011*, *LNCS 6906*: 149-163, 2011.