Design and Analysis of a Modified Remote Attestation Protocol

Monika Santra and Sateesh K. Peddoju
Dept. of Computer Sc. & Engg.
Indian Institute of Technology Roorkee
Roorkee, India - 247667.
Email: monikasantra@gmail.com, drpskfec@iitr.ac.in

A.K. Bhattacharjee and Arindam Khan
Reactor Control Division
Bhabha Atomic Research Centre
Mumbai, India
Email: {anup, arindamk}@barc.gov.in

***Abstract*—Secure interaction amongst system components is inherent to ensure the trustworthiness of the applications. In a distributed system, the attester should know whether the communicating client as well as the user who is using that client system is trustworthy. This can be achieved by a technique called remote attestation. This study presents a three-phase remote attestation protocol which provides relaxation over the low extensibility and low openness problem of binary remote attestation method, using the support of the SELinux module. It also analyses the performance of different existing and related binary remote attestation methods with the proposed approach which shows that the suggested remote attestation method is efficient. The results of the formal analysis are achieved using ProVerif tool which proves that the proposed remote attestation protocol satisfies several security properties such as secrecy, authenticity, indistinguishability and unlinkability.**

***Keywords*-Trusted computing; remote attestation; formal model analysis; attestation protocols; security analysis;**

I. Introduction

In the world of system security, maintaining the trust between two remote end parties is one of the major concerns. Secure communication mechanisms are needed to maintain the trust relation between two parties. In traditional system security, many techniques exist for the prevention of external invasion i.e., the attacks where adversary tries to gather information about the system by analyzing its behavior and then use it against the system. Some of the existing techniques to prevent of this kind of attacks are encryption, key establishment protocols etc. But in the modern distributed computing world, one of the major threats to network security is internal invasion that arises from within the terminal system. Thus to have more comprehensive, robust and efficient system security, the precautions must start from the root itself, i.e., by maintaining terminal or end point system security.

The main idea of trusted computing is to detect the unintended or malicious acts from the end parties using well-defined policies and then take measures accordingly. This is achieved by remote attestation. Remote attestation is a method by which a host or a client machine authenticates itself to a remote host called server by using both software and hardware configurations. In this method, before allowing the nodes to interact with each other, it checks whether the systems fulfill the desired requirements by judging the state of that system. Remote attestation is one of the most famous features provided by Trusted Computing Group (TCG) [4], [5].

One of the significant contributions of TCG is Trusted Platform Module (TPM) [5], [6]. TPM is a trusted hardware used for remote attestation to provide the integrity of a system and guarantee the trustworthiness of remote parties. It is a security co-processor which is assumed to be tamper resistant. There are many well-known software vendors who use TPM for ensuring security to some of their applications, such as Microsoft's BitLocker, and HP's HPProtectTools.

Semantic remote attestation, binary or configure based remote attestation, property-based attestation etc are the various types of remote attestation techniques. Some of which are Binary based remote attestation technique that performs attestation on the basis of raw or binary configurations of the system. So, by using TPM's integrity measurement architecture and binary remote attestation Protocol, the establishment of trust relation between two end parties is possible to achieve. This study focuses on binary based remote attestation protocols such as TCG-Integrity Reporting Protocol (TCG-IRP) [7], Robust-Integrity Reporting Protocol (Robust-IRP) [8], Stumpf method [9] and TLS-DAA protocol [10]. Most of these binary remote attestation protocols still use Trusted Third Party (TTP). Some problems still persist with binary remote attestation protocols, low extensibility and low openness being the important ones. The system update is restricted because of the fixed configuration in binary remote attestation. In other words, there is an update problem, which none of the above binary remote attestation protocols resolve.

In this study, a three-phase protocol, with the integration of SELinux [1] and TLS/SSL session [12], [13], is proposed. SELinux or the Security-Enhanced Linux [1] is a modified Linux secure kernel module with several features such as support for policy changes, and separate measures for protecting system integrity. The three phases of the protocol are Enrollment phase (phase I), Verification phase (phase II), and Update phase (phase III). In the enrollment phase, the client boots up for the first time and exports all its reference measurements to be stored on the server for further

2324-9013/17 $31.00 © 2017 IEEE
DOI 10.1109/Trustcom/BigDataSE/ICESS.2017.287

verification. In the second phase, i.e., when the system is not booting up for the first time, it verifies itself using the previous state measurement of the system. In the update phase, various changes done by the client system are updated on the server, thus solving the update issue of the binary remote attestation protocol.

To evaluate the proposed protocol, a formal analysis model has been performed in this study. This has been done by using an automatic security verification tool named *ProVerif* [2], [3], a popular tool to check or verify security properties of any security protocol.

The main contributions of this study are as follows:

- The proposed remote attestation protocol binds the system measurements with the server signed certificates in a modified way. It can later be used by the client systems, when booting for the 2nd time and onwards, to authenticate themselves with the server [Discussed in Section III].
- It can enforce policies to make the environment more suitable according to network need [Discussed in Section III].
- It does not allow any malicious or unintended changes during verification of the system by using various policies and Enforcement mode of SELinux [Discussed in Section IV].
- It finds a relaxation over binary remote attestation protocol by providing an update phase to the protocol [Discussed in Section III].
- It identifies a large class of binary remote attestation protocols and compares their security properties and time measurements [Discussed in Section V].
- It verifies the proposed protocol using ProVerif [2], [3]. It can satisfy various trace properties and privacy properties which are discussed in detail in section V.

Rest of the paper is organised as follows. The section II of this paper analyses related work regarding the binary remote attestation approaches. Section III and Section IV depict the proposed model and its security analysis respectively. Section V shows the performance analysis of various binary remote attestation approaches and formal model analysis of the proposed protocol using *ProVerif*. Finally, Section VI provides the conclusion and future work.

II. Related Work

This section provides a background related to the remote attestation approach and then describes the work achieved so far by various researchers in this binary attestation field. Before going directly to the related work, this section briefly discusses Trusted Platform Module (TPM) [5] and Security Enhanced Linux (SELinux) [1].

Trusted Platform Module (TPM) [5], also known as the root of trust, is a tamper-resistant security co-processor. TPM implements several primitive cryptographic functions such as public key cryptography, hash measurements etc. Security-Enhanced Linux [1] or SELinux can run in three modes a) Labeling mode, b) Permissive mode, and c) Enforcing mode. Each mode has its different abilities. In the labeling mode, all the system measurements can be taken for further verification. In permissive mode, if any application does not meet the security requirements, then a log is generated and stored. However, the application is simultaneously allowed to run. In Enforcing mode, if any application does not meet the security requirements, then a log is generated as well as blocking of the application is done. The booting mechanism of SELinux combined with TPM device ensures proper booting of the kernel. Therefore, it is the base of the trust system.

A lot of research has been done so far regarding the binary remote attestation protocol. In this study, a few existing binary remote attestation protocols are taken to compare, TCG-Integrity Reporting Protocol (TCG-IRP) [7], Robust-Integrity Reporting Protocol (Robust-IRP) [8], Stumpf method [9], TLS-DAA protocol [10], with the proposed remote attestation approach.

TCG-IRP [7] is the basic remote attestation protocol which provides integrity and authenticity on the basis of TPM functions, PCR values, and Stored Measurement Log (SML) values. By comparing the PCR hash values and log entries of SML values, the protocol attests the client machine. It is resistant to replay attacks and tampering but attacker bypasses the remote attestation protocol security using an honest client's credentials. Hence, this TCG-IRP doesn't prevent masquerading attacks.

Robust-IRP [8] proposes to overcome the weaknesses of TCG-IPR. This approach extends the TCG-IPR by implementing session creation procedure. This protocol uses Diffie-Hellman key establishment for maintaining the session. This key establishment procedure or session creation procedure makes sure that this protocol can prevent the masquerade attacks.

In [9], three approaches are suggested to improve the scalability of any platform when multiple simultaneous attestation requests arrive. These protocols work on batches of requests. All these protocols involve TTP for generating random numbers and synchronization tokens. So, Stumpf method [9] includes TTP overhead. Among these protocols, Multiple-Hash Attestation (MHA) is the slowest because of the ring buffer rotation time. Whenever one request comes, it goes into the ring buffer and it waits until the previous attestation ends. For the other attestation mechanisms, i.e. Timestamped Attestation and Tickstamp Attestation, TTP generates tokens and if tokens are ready, any system can go for attestation. All the these protocols establish sessions. Though this mechanism reduces average response time, it still includes TTP.

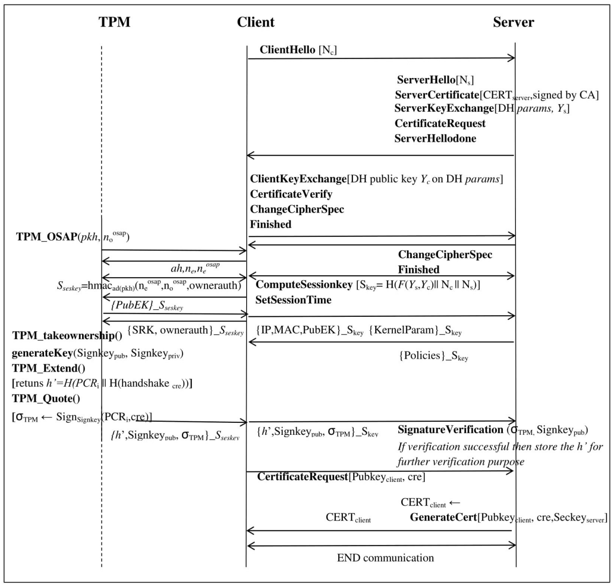
Direct Anonymous Attestation (DAA) [11] protocol consists of five phases i.e. Setup, Join, Sign, Verify and Link. In the Setup phase, issuer sets all its system parameters and publishes its public key and a rogue list to identify all the defected systems. Using the Join phase, a signer generates its own DAA credential created on its TPM secret key and later stored in the TPM. Using Sign phase, a signer generates a DAA signature. The verifier sends a nonce to maintain the freshness and also a string called basename (bsn) to control the user linkability. The Link phase checks whether two anonymous signatures are from the same signer or not. This Protocol is able to stop replay attack. However, because of its anonymous nature, it is prone to masquerade attack.

The next protocol [10] is an anonymous protocol based on TLS-DAA scheme. It is based on two main schemes, DAA and TLS(Transport Layer Security). TLS provides confidentiality, integrity, and availability. This protocol consists of three phases. In Negotiation Phase, client and server exchanges authentication information by Hello message extensions. In Platform Attestation and Key Exchange Phase, all the evidences, certificates, and keys are exchanged and verification of all the evidences are performed. And, in Key Derivation Phase, both the client and the server computes the session key, that is the Master Key and finishes the handshake protocol. This protocol satisfies the following requirements: Anonymity, User-control linkability, Unclonability, Anti-Replay attack and Anti-Masquerade attack. It also provides a secure channel for information exchange.

In contrast to all the above protocols, in this paper, the proposed remote attestation protocol combines remote attestation method with the enforcement techniques of SELinux. In this approach, the requirement specific policy enforcements are performed to provide better control over the attestation environment. It modifies the way of binding the system authenticity with the server signed X.509 certificate for attestation. The verification phase (phase II) prevents the attacks from any unintended or malicious changes with the help of SELinux's enforcement policies. This study also provides a way to recover from one of the most important deficiencies of Binary Attestation mechanism i.e., the update problem by relaxing the remote attestation approach. It allows the update option with the admin acknowledgment in its implementation.

III. Proposed Model

This section describes the proposed three-phase remote attestation model consisting of three phases i.e., Enrollment phase (Phase I), Verification phase (Phase II), and Update phase (Phase III). In the Enrollment phase, client machine's boot_setup has been done. When the system is booting up for the very first time, this boot_setup is performed in the Enrollment phase to initialize the framework of that system. This boot_setup is done only once. Verification phase will run whenever client wants to access the service from the server or the server challenges the client. Accordingly, if the system gets updated, then the update phase reference measurements can be performed. Detailed description of the three phases are as follows:



TPM
Client
Server
ClientHello [N_c]
ServerHello[N_s]
ServerCertificate[CERT_server,signed by CA]
ServerKeyExchange[DH params, Y_s]
CertificateRequest
ServerHellodone
ClientKeyExchange[DH public key Y_c on DH params]
CertificateVerify
ChangeCipherSpec
Finished
TPM_OSAP(pkh, n_o^osap)
ah,n_e,n_e^osap
ChangeCipherSpec
Finished
S_seskey=hmac_ad(pkh)(n_e^osap,n_o^osap,ownerauth)
ComputeSessionkey [S_key= H(F(Y_s,Y_c)|| N_c || N_s)]
SetSessionTime
{PubEK}_S_seskey
{SRK, ownerauth}_S_seskey
{IP,MAC,PubEK}_S_key {KernelParam}_S_key
TPM_takeownership()
generateKey(Signkey_pub, Signkey_priv)
{Policies}_S_key
TPM_Extend()
[retuns h'=H(PCR_i || H(handshake _cre))]
TPM_Quote()
[σ_TPM ← Sign_Signkey(PCR_i,cre)]
{h',Signkey_pub, σ_TPM}_S_seskey
{h',Signkey_pub, σ_TPM}_S_key
SignatureVerification (σ_TPM, Signkey_pub)
If verification successful then store the h' for further verification purpose
CertificateRequest[Pubkey_client, cre]
CERT_client ←
CERT_client
GenerateCert[Pubkey_client, cre,Seckey_server]
END communication

Figure 1. Enrolment phase of the Remote Attestation protocol.

Enrolment phase: This phase runs whenever client system is booting for the first time in the network. In this system, client runs in the labelling mode. Figure 1 shows the implementation of the Enrolment phase. For the communication between the server and the host, it uses TLS/SSL session to create a point to point communication channel, and for the session establishment between host and the TPM, it uses the TPM_OSAP sessions. The host needs to take ownership over the TPM using TPM_takeownership() command as it boots up for the first time. Later, client system generates reference measurements and exports them to the server. The server saves these reference measurements and use them for further verification process. These references are taken based on the policies. For identification purpose, the server assigns a Universally Unique Identifier (UUID) to each of the clients. After validating the signature, the server issues an X.509 certificate signed by the server, to the client for future reference. Enrollment phase is performed only once during the lifetime of the client system in that network. This phase may increase some overhead but it makes sure that the proposed protocol is Third Party free and it runs only once.

Verification Phase: The server asks the client to send the references it needs, according to the enforcing mode policies. According to the policies, client machine performs the TPM_Extend() to generate hash and TPM_Quote() to get the signature and public signing key. The server verifies the signature after getting these values. If the signature is

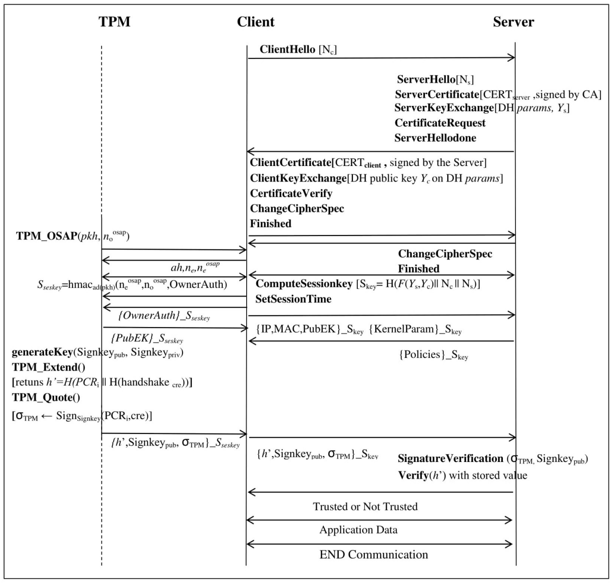

TPM
Client
Server
ClientHello $[N_c]$
ServerHello$[N_s]$
ServerCertificate[CERT_{server} ,signed by CA]
ServerKeyExchange[DH $params$, Y_s]
CertificateRequest
ServerHellodone
ClientCertificate[$\text{CERT}_{\textbf{client}}$, signed by the Server]
ClientKeyExchange[DH public key Y_c on DH $params$]
CertificateVerify
ChangeCipherSpec
Finished
TPM_OSAP(pkh, n_o^{osap})
ah, n_e, n_e^{osap}
S_{seskey}=hmac$_{ad(pkh)}$(n_e^{osap},n_o^{osap},OwnerAuth)
ChangeCipherSpec
Finished
ComputeSessionkey [S_{key}= H($F(Y_s,Y_c)$|| N_c || N_s)]
SetSessionTime
{OwnerAuth}_S_{seskey}
{PubEK}_S_{seskey}
{IP,MAC,PubEK}_S_{key} {KernelParam}_S_{key}
{Policies}_S_{key}
generateKey(Signkey$_{pub}$, Signkey$_{priv}$)
TPM_Extend()
[retuns $h'=H(PCR_i$ || H(handshake $_{cre}$))]
TPM_Quote()
[$\sigma_{TPM} \leftarrow$ Sign$_{Signkey}$(PCR$_i$,cre)]
{h',Signkey$_{pub}$, σ_{TPM}}_S_{seskey}
{h',Signkey$_{pub}$, σ_{TPM}}_S_{kev}
SignatureVerification ($\sigma_{TPM,}$ Signkey$_{pub}$)
Verify(h') with stored value
Trusted or Not Trusted
Application Data
END Communication

Figure 2. Verification phase of the Remote Attestation protocol.

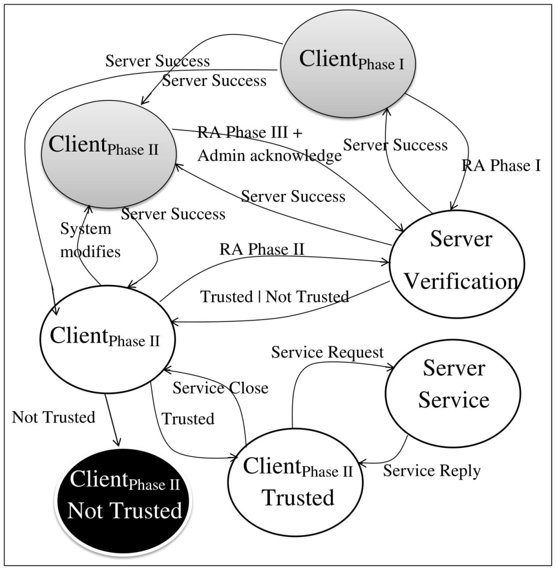


Figure 3. State Transition diagram of the proposed Remote Attestation approach.

valid, then according to the stored measurements, the server decides whether the system is verified or not. Here, the client system runs in the enforcing mode of SELinux. In Phase II, as shown in Figure 2, during the TLS/SSL session establishment and mutual authentication, both client and server side certificate verification is performed.

Update Phase: This phase is included to eliminate the update problem of binary attestation. In this phase, client machine needs to be in labeling mode. Any OS update or application update changes the manifest reference policies stored in the verifier or server. So, using the admin acknowledgment, the client machine is allowed to update the reference measurements. In this study, it is assumed that the superuser or admin of the client system is trusted. The detail of this phase is almost similar to Phase II, which is depicted in Figure 2. The basic difference between Phase III from phase II is that after the session creation and credential exchange, client system sends names of the packages that need to be change, in the server with the admin acknowledgment. If the signature is verified again, the server updates only those modified packages in the database. Here too, the client system needs to be in labeling mode.

A nonce communication follows each message sent and received by both the parties. The proposed protocol also maintains a session timer with different purposes for different phases. For example, in Phase I and Phase III, if session timeout occurs, then both parties establishes a new session key based on the previously shared parameters. But, for Phase II of the protocol, if session timeout occurs, then the protocol aborts and the system needs to try again to verify itself.

The figure 3 shows the state transition diagram of remote attestation protocol. The states that are in gray are the client nodes with SELinux being in labeling mode and all the other client states are in enforcing mode. This study assumes that the server or the attester is trusted and the server system works in any mode of SELinux. The black client state is a dead state. The client enters into this state when it receives Not Trusted reply from the server. So, whenever the client machine changes, it needs to go to labeling mode for updating the system. If some malicious file gets updated, the admin comes to know because Phase III needs admin acknowledgments for changing the server reference measurements.

IV. Security Analysis

In this section, analysis of the proposed remote attestation protocol is performed, considering the security of the cryptographic primitives such as SHA-1 and AES. This section also defines some of the common attack scenarios and corresponding mitigation techniques that are applicable in any remote attestation environment. The following are the attacks analyzed in this study.

1) Unintended or malicious modification of Files: This attack shows all the scenarios where a malicious party tries to access the system or tries to modify the file system to get the information about that system. In case of the proposed protocol, modification of the files is only allowed in labeling mode. The enforcing mode detects all the malicious modification of files.

2) Dictionary attack and Replay Attack Against Remote Attestation: The adversary tries to use some messages of the client to create another session with the server so that it can present itself to the server as an actual authenticated client. The proposed protocol uses a 160-bit nonce to avoid such kind of attacks. Since the nonce value changes every time, the attacker fails to use previous session messages in the current sessions.

3) Memory Copy Attack: The attacker tries to make fake page table in the client system and uses that fake page table to forcefully create a copy of its modified remote attestation module to the address of the genuine remote attestation module. Without compromising the OS kernel, it is impossible to achieve this kind of attack. In the monitoring of integrity measurement system, compromising the OS kernel is hard to achieve.

4) Man in the Middle Attack: In this attack, the attacker actively eavesdrops on the system by creating an illusion that the client and the server are communicating with each other. The proposed protocol maintains TLS/SSL session between the client and the server.It prevents the Man in the Middle attack using this authenticated channel between the client and the server.

5) Timing Attacks: Timing attacks are also known as side channel attacks. Here, the adversary tries to figure out the cryptographic algorithm using its execution time. To avoid this kind of attack scenario, time randomization is used by introducing extra no-operation (NOP).

V. Evaluation

The evaluation is divided into two sections. The first section depicts the comparison of proposed approach with other related remote attestation processes, in terms of time, memory and other aspects. The second section shows verification of the cryptographic model using ProVerif [2], [3] tool.

A. Performance Evaluation

The proposed remote attestation protocol and a few related protocols [7], [8], [9], [10], discussed in Section II, are compared. These related protocols are TCG-IRP [7], Robust-IRP [8], Stumpf method [9], and TLS-DAA protocol [10]. This evaluation consists of a table which shows the performance comparison in terms of cryptographic notations. This study also shows the comparison of all the protocols from various aspects such as average response time and communication load. In these experiments, all the measurements are taken on an x86_64 machine with intel core i5 @2.30GHz and 8GB RAM.

All the protocols are independent and satisfies the requirements to provide simultaneous verification of different clients. Table I compares the average Response time, Minimum Response time, Maximum Response time and communication load of different protocols needed during the attestation procedure. In all the above protocols, the system needs to communicate with the hardware device i.e. TPM, for performing the signature generation i.e. the Quote operation. This communication between hardware and software consumes a longer time than other server-client communications. So, ignoring the other less time consuming parameters, the response time of the protocols [7], [8], [9], [10] and the proposed protocol are given in terms of the quote operation time. The performance comparison is shown in Table I.

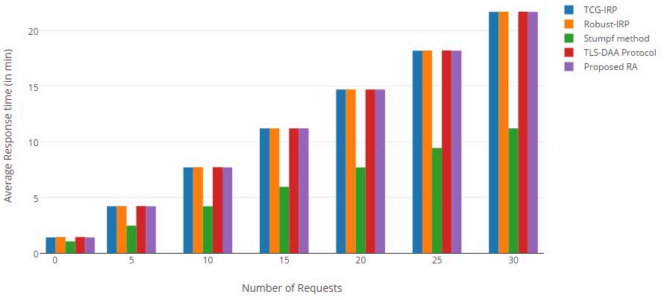


Figure 4. Average Response time vs number of requests.

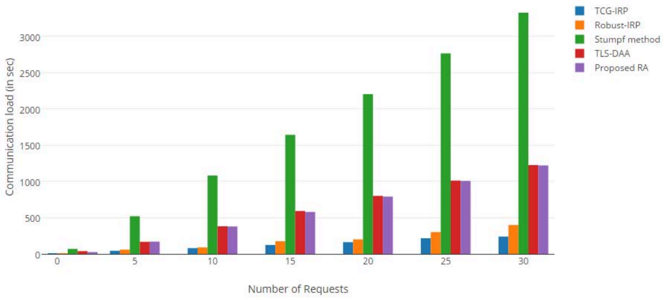


Figure 5. Communication load vs number of requests.

Table I
Comparison of Proposed RA protocol with other similar models.

Protocol	Avg. Response	Min.Responce	Max.Response	Communication Load
TGC-IRP [7]	$\frac{n_t+1}{2}t_q$	t_q	$n_t t_q$	$n_t c$
Robust-IRP [8]	$\frac{n_t+1}{2}t_q$	t_q	$n_t t_q$	$n_t(c+t_s)$
Stumpf Method [9]	$\frac{n_t+n_b}{n_b^2}t_q$	t_q	$\frac{n_t}{n_b}t_q$	$n_t n_b(c+t_{ts})$
TLS-DAA Protocol [10]	$\frac{n_t+1}{2}t_q$	t_q	$n_t t_q$	$n_t(c+t_s+t_{DAA})$
Remote Attestation (Proposed approach)	$\frac{n_t+1}{2}t_q$	t_q	$n_t t_q$	$n_t(c+t_s+t_{cert})$

In the table, n_t is the total number of requests coming to the server, n_b is the number of requests in one batch in Stumpf method [9], t_q is the time needed for quote operation performing, c is the length of the nonce, t_s is the session performing time, t_{ts} is the tick session time in case of Stumpf method [9], t_{DAA} is the time needed for DAA compilation and t_{cert} is the certificate management time. Taking t_q as 1.3 min on average and n_b as 2 i.e. each batch containing 2 requests only and varying n_t, average response time and communication load graphs are depicted in Figure 4 and Figure 5 respectively.

From Figure 4, it is observed that most of the compared protocols do not deal with batch processing except for Stumpf method [9], while all of them take more or less

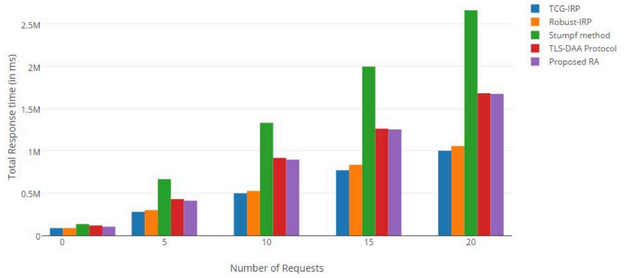


Figure 6. Total response time or measured latency graph vs number of requests.

the same response time with respect to the Quote operation. This is not the case for Stumpf method [9] due to batch processing. From Figure 5, it is observed that the proposed protocol has slightly less communication load than TLS-DAA because the suggested protocol does not deal with five phases of DAA protocol which takes more time than the certificate management time.

The implementation of all the above-compared protocols is very efficient and configurable. The results shown are the average of different independent iterations of the attestation procedure. Measurement of latency is done for the full attestation process i.e. from the time of attestation request arrival to the server to the time when the last message was sent from the server to the client. The latency comparison graph is depicted in Figure 6.

From Figure 6, it is observed that the simpler protocols i.e. TCG-IRP [7] and Robust-IRP [8], take less response time. As the communication load is an important part of the total response time, the proposed protocol takes a few milliseconds less than the TLS-DAA protocol. Due to high TTP overhead, Stumpf method [9] takes the highest total response time.

In the proposed remote attestation approach, measurements for both the phases i.e phase I and phase II are taken. As phase I occurs only once during the first boot process, phase II latency timings are the verification time of the protocol. During phase I, the proposed remote attestation protocol takes around 5.2% CPU and 5092 KB memory space for one attestation procedure. For phase II, i.e., for verification, it takes around 3.5% CPU and 4479.2 KB memory space.

During phase II, varying the accepted request number, the server side memory analysis graph is depicted in Figure 7.

It can be observed from the memory analysis graph that as the number of requests increases, the consumption of memory increases multiplicatively, i.e., if memory consumption is m units for one request then for n number of requests, the memory needed is $m \times n$ units. Figure 8 shows the CPU usage analysis graph of the proposed protocol. In this study, the comparison is done for both Phase I and Phase II of the protocol. From Figure 8, it is observed that as the number of requests increases, the CPU usage also increases, but due to the presence of multicore and parallel threading technologies, the whole process gets faster and the CPU usage is not increased multiplicatively as the memory usage.

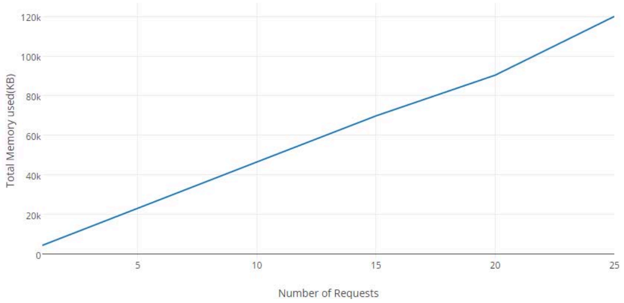


Figure 7. Memory analysis graph.

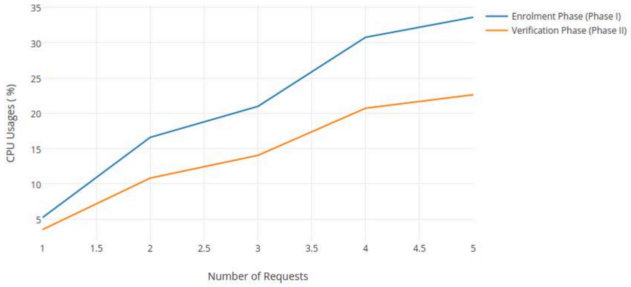


Figure 8. CPU analysis graph.

B. Model verification

In this section, a detailed formal analysis of the suggested remote attestation approach is presented. The formal analysis is achieved by setting different assertion properties given by *ProVerif* [2], [3]. ProVerif is a tool used for automatic analyzing and security property verification of any security related protocols based on variation of applied pi-calculus [14], [15]. ProVerif also allows to define user-specific cryptographic primitives. It also successfully proves the robustness and safety of any protocol by finding counter-examples of attack scenarios. Using ProVerif supported secrecy, correspondence property, and observational equivalence property, many security related properties can be proven for the proposed remote attestation protocol [3]. For all the above reasons, ProVerif [2], [3] tool is chosen to analyze and verify the proposed protocol. For verifying security properties, various ProVerif defined events and assertions are used.

Several trace and privacy properties are verified using this tool. Secrecy or confidentiality, and authentication are called trace properties [3] as the verification of the property is given by ensuring that there is no trace of any attack counter-example over the protocol. However, Frame Opacity, Well-Authentication and unlinkability are called privacy properties [16], [17], [18] as the verification of these properties depends upon the indistinguishability of different sessions of the same protocol. The satisfied properties are discussed in detail in the following parts.

Secrecy: Secrecy or confidentiality property of data is defined by the inaccessibility of that data by the adversary. By querying that data in the model, it is checked and confirmed that the data is not being leaked to the attacker i.e., the attacker does not have access to that data. If the above condition is satisfied, then the secrecy property is proved. One of the examples of checking the property of the owner authorisation data is as follows:

Query attacker(OwnerAuth);
The corresponding result of this query is following:
RESULT not attacker (OwnerAuth [ ]) is true.

This means that the secrecy property holds. This study has also performed the same secrecy property for different data and all of them hold the property.

Authentication: The authentication property is proved by the correspondence property. The correspondence property defines that "$X \rightarrow Y$" i.e., if X happens then Y occurs before X. For using this property, two events are defined in this model, i.e. HostSuccess and ServerSucess, and execute the following query on it:

Query x:key ; inj-event (ServerSuccess(x)) → inj-event (HostSuccess(x));

This query signifies that if any server verifies any signature, then this signature must be signed by some verified client system. The result of the above query is following:

RESULT inj-event (ServerSuccess(x)) → inj-event (HostSuccess(x)) is true.

As the result of this query is true in this approach, that means the proposed attestation system verifies the correspondence or authentication property successfully.

Frame Opacity: The next security property satisfied by this protocol is Frame Opacity [16], [17]. It suggests that different outputs should maintain indistinguishability and pure randomness i.e. these outputs should not reveal anything about the system to the attacker. In the formal model, indistinguishability is defined by observational equivalence. The definition of observational equivalence is that the attacker should not be able to differentiate between two processes P and Q. This observational equivalence is achieved by comparing the current frame with an ideal frame, i.e., by changing the frame values with some new names and then check whether the attacker detects the difference or not. This model satisfies the observational equivalence i.e. Frame Opacity [16], [17] is also satisfied by the proposed protocol model.

Well Authentication: The well authentication condition prevents an attacker to get information from the outcomes of the conditional statements. So, it is needed to make sure that all the interactions are intended and honest. For that, it is required to maintain an event for the sending and receiving of each of the messages and then check to see whether the execution order is actually maintained or not. The example query for checking Well-Authentication is as follows:

query k:key, b:bitstring;

(event(Ctest_1(k))→(event(Cin_2(k))→(event(Sout_2(k))
→(event(Sin_1(b))→(event(Cout_1(b))→(event(Cin_1(b))
→(event(Sout_1(b)))))))));

As the output of this query is also true, the model satisfies the Well-Authentication condition.

Unlinkability: Unlinkability [18] means that the adversary should not be able to use any information from the same process'different sessions against the system and it should not be able to differentiate between two sessions of the same process. There should be no difference between a system with an arbitrary number of process repetitions and the system which actually runs once. This definition is also called strong unlinkability. There is no direct way to check this property using the ProVerif auto verification tool, but it can be indirectly proven. According to the work done by [16] and [17], it is proven that if any security protocol satisfies Frame Opacity [16], [17] and Well-Authentication property [16], [17], then that protocol also satisfies the unlinkability property. So, as the suggested protocol satisfied both the Frame Opacity and Well-Authentication property, it can be deduced that it also satisfies the unlinkability property.

VI. Conclusion and Future Work

In this study, a three-phase binary remote attestation protocol is proposed by using the existing SELinux [1] kernel module policy integration and TLS/SSL authentication mechanism [12], [13]. This protocol can give relaxation over the low openness or update problem of binary remote attestation. Moreover, performance evaluation on various related binary remote attestation protocols and security analysis on the proposed protocols is performed. As a case study, the formal analysis of the proposed remote attestation protocol is performed. The results clearly indicate that the proposed protocol satisfies various security related properties such as unreachability, authenticity, indistinguishability, and well-authentication. By combining the last two properties, it is shown that this protocol is resistant to linkability i.e. the proposed protocol supports unlinkability property and assures its robustness. Since the proposed protocol is still a binary remote attestation protocol and is based on PCR measurements, a few other problems also exist [19]. The future work includes converting this binary attestation protocol to property-based attestation by creating formal models which can efficiently perform the conversion between the configurations and the properties.

Acknowledgment

This research work has been supported by Bhabha Atomic Research Centre, Mumbai, India.

References

[1] Lawrence, Steve (2016-02-23). "Release 2016-02-23". SELinux. SELinux Project. Retrieved 2016-02-24.

[2] B. Blanchet, "An efficient cryptographic protocol verifier based on prolog rules," Proceedings. 14th IEEE Computer Security Foundations Workshop, 2001., 2001, pp. 82-96.

[3] Bruno Blanchet.ProVerif: Automatic Cryptographic Protocol Verifier User Manual, 2008

[4] Grawrock D. TCG Specification Architecture Overview Revision 1.4[J]. 2007.10.01.

[5] Trusted Computing Group: TCG TPM specification 2.0 (2012)

[6] "ISO/IEC 11889-1: Trusted platform module library - Part 1: Architecture", ISO Standards Catalogue, August 2015

[7] Reiner Sailer, Xiaolan Zhang, Trent Jaeger, and Leendert van Doorn. 2004. "Design and implementation of a TCG-based integrity measurement architecture". In Proceedings of the 13th conference on USENIX Security Symposium - Volume 13 (SSYM'04), Vol. 13. USENIX Association, Berkeley, CA, USA, 16-16.

[8] F. Stumpf, O. Tafreschi, P. Roder, and C. Eckert. ARobust Integrity Reporting Protocol for RemoteAttestation. InSecond Workshop on Advances inTrusted Computing (WATC'06 Fall), Tokyo, Japan, November 2006.

[9] Frederic Stumpf, Andreas Fuchs, Stefan Katzenbeisser, and Claudia Eckert. 2008. Improving the scalability of platform attestation. In Proceedings of the 3rd ACM workshop on Scalable trusted computing (STC '08). ACM, New York, NY, USA, 1-10.

[10] A. Lan, Z. Han, D. Zhang, Y. Jiang, T. Liu and M. Li, "An Anonymous Remote Attestation Protocol to Prevent Masquerading Attack," 2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops, Bali, 2014, pp. 590-595.

[11] Ernie Brickell, Jan Camenisch, Liqun Chen, Direct Anonymous Attestation, CCS '04Proceedings of the 11th ACM conference on Computer and communications security,2004

[12] Y. Yu, H. Sun and Y. Kong, "Expand the SSL/TLS protocol on Trusted Platform Module," 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), Taiyuan, 2010, pp. V11-48-V11-51.

[13] Frederik Armknecht, Yacine Gasmi, Ahmad-Reza Sadeghi, Patrick Stewin, Martin Unger, Gianluca Ramunno, and Davide Vernizzi. 2008. An efficient implementation of trusted channels based on openssl. In Proceedings of the 3rd ACM workshop on Scalable trusted computing (STC '08). ACM, New York, NY, USA, 41-50.

[14] M. Backes, M. Maffei, and D. Unruh, Zero-knowledge in the applied pi-calculus and automated verification of the direct anonymous attestation protocol, in Security and Privacy,2008. SP 2008. IEEE Symposium on. IEEE, 2008, pp. 202215.

[15] M. D. Ryan and B. Smyth, Applied pi calculus, 2011.

[16] L. Hirschi, D. Baelde and S. Delaune, "A Method for Verifying Privacy-Type Properties: The Unbounded Case," 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, 2016, pp. 564-581.

[17] M. Arapinis, T. Chothia, E. Ritter and M. Ryan, "Analysing Unlinkability and Anonymity Using the Applied Pi Calculus," 2010 23rd IEEE Computer Security Foundations Symposium, Edinburgh, 2010, pp. 107-121.

[18] Iso 15408-2: Common criteria for information technology security evaluation - part 2: Security functional components, ISO Standards Catalogue, July 2009.

[19] Ahmad-Reza Sadeghi and Christian Stble. 2004. Property-based attestation for computing platforms: caring about properties, not mechanisms. In Proceedings of the 2004 workshop on New security paradigms (NSPW '04). ACM, New York, NY, USA, 67-77.