Linking Remote Attestation to Secure Tunnel Endpoints

Kenneth Goldman
IBM T. J. Watson Research Center
19 Skyline Drive
Hawthorne NY 10532
kgoldman@us.ibm.com

Ronald Perez
IBM T. J. Watson Research Center
19 Skyline Drive
Hawthorne NY 10532
ronpz@us.ibm.com

Reiner Sailer
IBM T. J. Watson Research Center
19 Skyline Drive
Hawthorne NY 10532
sailer@us.ibm.com

ABSTRACT—Client-Server applications have become the backbone of the Internet and are processing increasingly sensitive information. We have come to rely on the correct behavior and trustworthiness of online banking, online shopping, and other remote access services. These services are implemented as cooperating processes on different platforms. To trust distributed services, one must trust each cooperating process and their interconnection.

Common practice today is to establish secure tunnels to protect the communication between local and remote processes. Typically, a user controls the local system. The user also controls the security of the tunnel through negotiation and authentication protocols. Ongoing and published work examines how to create and monitor properties of remote systems. What is missing is the link or binding between such properties and the actual remote tunnel endpoint.

We examine here how to link specific properties of a remote system – gained through TPM-based attestation– to secure tunnel endpoints to counter attacks where a compromised authenticated SSL endpoint relays the TPM-based attestation to another system. We show how the proposed mechanism can be deployed in virtualized environments to create inexpensive SSL endpoint certificates and instant revocation that scales Internet-wide.

Categories and Subject Descriptors
D.4.6[**Operating Systems**]:Security and Protection–Authentication

General Terms
Measurement, Security, Verification.

Keywords
Trusted Platform Module, Certificates.

1. INTRODUCTION
Applications must establish trust in remote computing services. This trust includes both determination of the remote endpoint identity and knowledge of the software running on the platform.

Endpoint identity is currently determined through protocols such as SSL [1] or IPSec [2], using public key signatures and certificates issued by trusted certificate authorities. These protocols establish a secure connection to a remote server using a known key.

STC'06, November 3, 2006, Alexandria, Virginia, USA.

Trust in the software platform can be established with the aid of remote attestation, a form of which is supported by emerging technologies defined by the Trusted Computing Group (TCG).

This paper addresses two problems associated with the aforementioned technologies:

- the lack of linkage between the endpoint identity determination and remote platform attestation
- the scalability problems when issuing and revoking server certificates [3] including frequently issuing and revoking certificates as well as distributing certificate revocation lists

The scalability problems are amplified in a virtual machine environment, where virtual servers are dynamically created and destroyed. Endpoint certificates must be issued and revoked dynamically for such virtual servers.

Section 2 provides a brief overview of existing and related work in this space. We present our scalable approach to linking security properties and remote tunnel endpoints in section 3 using the example of SSL. Section 4 explores the value of our mechanism in virtualized environments and how it can improve on the current PKI key revocation scalability problem.

2. STATE OF THE ART
Endpoint identity determination currently uses well established protocols such as SSL and IPSec. As a session is established, the client receives a certificate and challenges the server to prove possession of the associated private key. Validation includes:

- validating the challenge signature
- validating the certificate signature and the certificate chain
- validating that the static server endpoint properties are as expected

A typical static end point property included in key certificates is the server domain name. In this case, the tunnel authentication protocol assures the client that it is connected to the correct domain.

Remote server attestation uses TCG [4] technology, specifically the Trusted Platform Module (TPM) "quote" function. The quote creates a signature of the current platform software state. This state is reported though a log of software events, such as calling a higher software layer, starting a service, or reading a configuration file [5]. These events are recorded as "measurements", which are cryptographically protected by extending them into Platform Configuration Registers (PCRs). Signing the PCRs effectively signs the event log.

The signing Attestation Identity Key (AIK) used in the quote obtains a certificate signed by a Privacy CA. That certificate attests to the trust properties of the platform. The AIK is generated on and remains locked to the TPM, which is itself physically attached to the platform.

We use the TCG/TPM attacker model, which does not include hardware attacks on the TPM, so the AIK cannot be moved or copied.

Remote attestation allows the client to make a decision about the trust state of the server. The server cannot misrepresent its configuration without detection. The platform state includes the hardware platform, boot code such as firmware or BIOS, the operating system and applications.

The client in this example validates two signatures (the TPM quote and the network challenge) and walks two certificate chains (the AIK certificate and the SSL certificate).

Previous research [10] and ongoing work within TCG (TNC-SG [6]) examine how TPM-based attestation can be used to establish client properties before permitting a local or remote client access to centralized services. Both approaches share goals similar to ours and could thus benefit from the proposals presented here.

Prior work in the trusted computing field has proposed TPM-based attestation of system properties as a way to address scalability issues which arise when otherwise having to attest to every program load or parameter / configuration change which might affect endpoint security ([5],[7],). However, to our knowledge no existing work describes how to effectively connect established or derived specific security-related properties to higher layer secure tunnel endpoints.

3. LINKING REMOTE ATTESTATION TO SECURE TUNNEL ENDPOINTS

In this section, we examine how to securely couple properties of TPM-based attestation to secure tunnel endpoints by analyzing the problems with existing systems.

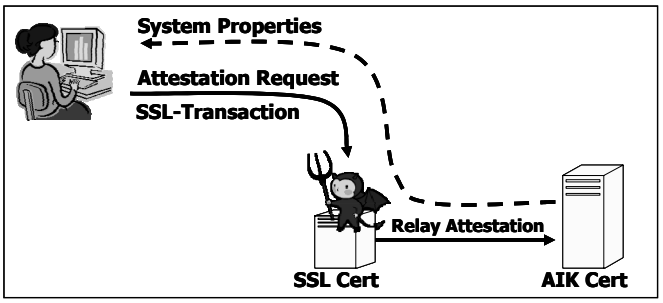

Figure 1: Disconnected SSL Tunnel and Attestation Endpoints

3.1 Problem A – No Link

Although the client establishes SSL endpoint identity and platform trust, there is no linkage between the two. That is, the two parts may come from different servers (cf. Fig 1).

For example, an untrusted SSL server might correctly demonstrate its identity. However, it might relay the attestation challenge to another, trusted server, see Figure 1. The client cannot detect this relay attack even if the attestation protocol is activated through the SSL tunnel.

3.2 Solution A – Creating a Link

To create this link and foil the relay attack, we propose adding a measurement of the endpoint static properties to the TPM event log and PCRs. An example property is the SSL public key or certificate (cf. Fig 2).

As before, the client validates the SSL certificate chain and the AIK certificate chain. As before, the client now has trust in the server endpoint and the platform hardware and software state.

In addition, the event log now connects the endpoint identity to the properties of the underlying platform. That is, the remote attestation quote of the platform properties is linked to the SSL certificate and thus the domain name.

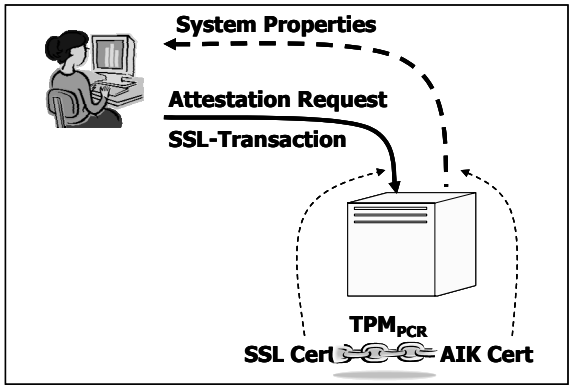

Figure 2: Measuring the SSL Endpoint Certificate

A compromised server cannot relay the attestation request without detection. The relayed attestation will include a measurement of its own endpoint properties, not those of the compromised server, in its event log. The client can now differentiate between the SSL endpoints.

The next subsection discusses the problem of compromised SSL private keys, which could be used on a compromised system to masquerade the legitimate SSL private key owner.

3.3 Problem B – A Compromised SSL Endpoint Private Key

There is currently no graceful way of handling a compromised SSL endpoint private key. Whoever has the private key can impersonate the platform.

In theory, certificate authorities maintain a certificate revocation list (CRL) with a list of invalid certificates. Clients check this list before using a certificate.

In practice, there are several problems with this approach [3]:

- Clients such as web browsers do not check CRLs.
- The solution does not scale as CRLs grow.
- The infrastructure is susceptible to denial-of-service attacks on the CRL server.

3.4 Solution B – Link AIK and SSL Endpoint Key

To create a link between the AIK and the server SSL endpoint key, we propose a third "Platform Property" certificate (cf. Figure 3). The Platform Property certificate contains:

- endpoint properties, as in the SSL endpoint certificate
- the AIK public key, as in the AIK certificate
- a signature by a CA

The endpoint properties include information such as the domain name and the organization. It does not include the actual SSL endpoint public key. Therefore, the Platform Property certificate is less likely

to be revoked than the SSL endpoint certificate because the related AIK stays inside the hardware TPM and is very unlikely to be compromised. Re-issuing an endpoint certificate will not impact the Platform Property and AIK certificates.

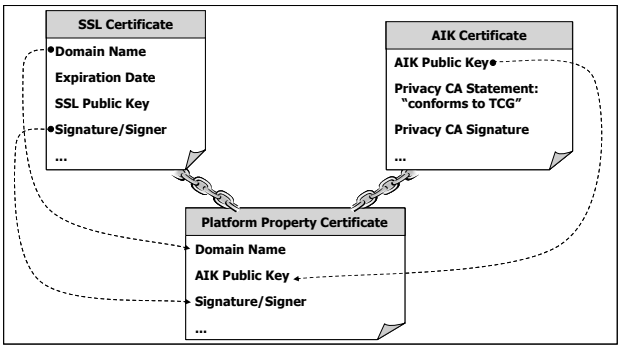

Figure 3: The Platform Property Certificate Securely Links SSL Endpoint and AIK Certificates

The CA in this case can be the same or different from the CA signing the endpoint certificate. Using the same CA speeds client verification, but the security properties are the same in either case.

This solution cryptographically links the platform and the SSL endpoint properties. That is, it connects the endpoint properties to the physical platform through the hardware TPM AIK.

The client validates:

- the SSL endpoint certificate chain
- the AIK certificate chain
- the Platform Property certificate chain

Again, there are two signatures, one generated by the TPM AIK and one generated by the SSL Endpoint.

If the attacker tries to use the compromised SSL endpoint private key, validation fails because either:

- the attacker uses its own Platform Property certificate, which does not match the compromised SSL endpoint certificate, or
- the attacker uses a Platform Property certificate matching the SSL endpoint certificate, which does not match the AIK the attacker uses in the quote

Therefore, there is no need to centrally revoke a compromised SSL endpoint key. It is of no use to the attacker. The attacker does not have access to the original AIK to create a bogus quote. Since the AIK is kept within the secure boundary of the TPM, compromise requires a physical attack on the platform.

3.5 Problem C – Three Certificates to Validate

While the Platform Property certificate solves the problem of a compromised SSL endpoint key, the client is burdened with validating three certificates. It must walk either 2 or 3 certificate chains, depending on whether the same or a different CA signs the Platform Property and SSL endpoint certificates.

3.6 Solution C – A Self Signed Endpoint Certificate

The SSL endpoint certificate adds no trust that is not already contained in the Platform Property and AIK certificates. It has SSL endpoint identity information signed by the CA, but Solution B already adds this data to the Platform Property certificate. It contains the SSL endpoint public key, but Solution A adds this public key or the related certificate to the PCR measurement list, signed by the AIK.

Therefore, we propose to make the SSL endpoint certificate a self signed certificate. That is, it does not need third party certification, since its contents are securely vouched for by the other certificates.

Besides the obvious performance improvement on the client side, validating two certificate chains rather than three, there are several other advantages to this approach:

- Tunnel endpoint keys can change and be set to expire often.
- When the endpoint public key changes, there is no need to purchase a new certificate.
- One can locally create different keys for SSL, IPSec, etc.
- One can create keys of different lengths for different applications, trading off security vs. performance as appropriate.

4. Virtualization

The solution described so far is quite valuable in virtualized systems. Here, user virtual machines or partitions are dynamically created and destroyed as the underlying hypervisor runs. A privileged partition containing the TPM support is instantiated at boot time and persists for the lifetime of the hypervisor. It provides TPM services to user partitions [8].

Suppose a user partition is instantiated and requires an SSL certificate to act as a web server. We propose that the virtual endpoint certificate is created by the TPM partition, with these properties:

- It contains the endpoint properties of the virtual server, such as the domain name and dynamic trust properties of the virtual partition.
- It is self signed.

Dynamic trust properties are those determined as a partition is running. Typically, a local authority running on a trusted partition monitors the virtual partitions [9],[10],[11]. The monitor must detect trust state changes to monitored properties and quickly revoke the virtual endpoint certificate when necessary.

Dynamic properties monitored include:

- comparison against a list of known vulnerabilities
- maintenance against a standard, such as a required patch list
- compliance to privacy standards or business guidelines
- events such as system compromise due to intrusions

At creation, this virtual endpoint certificate data is added to the event log and PCRs of the privileged partition, along with a notation that this is a valid certificate. As in the non-virtualized system, this binds the certificate to the AIK and thus to the trusted platform (cf. Fig. 4).

The Platform Property certificate in this case defines a static physical endpoint for the platform, not a dynamically created endpoint. The client validates:

- the quote and AIK certificate, establishing trust in the physical platform and software through the hypervisor and TPM partition
- the Platform Property certificate connecting the AIK to the platform running the hypervisor
- the self signed endpoint certificate, which it validates not through a certificate chain but by its presence in the attested event log and PCRs

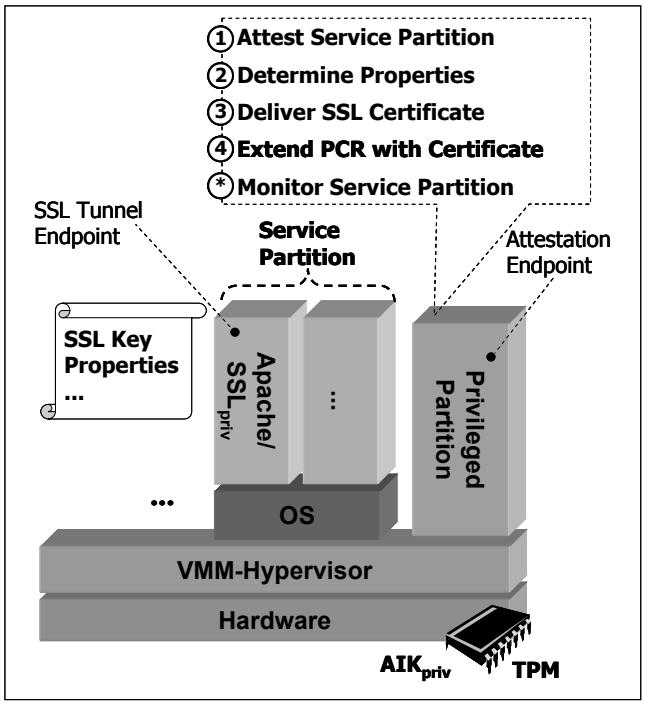

Figure 4: Privileged TPM Partition as On-line Certification Authority for Service Partition Certificates

A partition cannot create its own self signed endpoint certificate, because it has no way of adding it to the event log and PCRs. So the remote party is assured that the certificate was created by the trusted TPM partition.

Revocation of Virtual Endpoint Certificates

When a partition is destroyed or the dynamic properties of the partition change, its endpoint certificate must be revoked. This is now easily done by adding the certificate to the event log and PCRs again, this time with a notation that it is an invalid certificate. This is much simpler than maintaining an ever growing CRL as virtual servers are created and destroyed in an active data center.

Revocation is easy and quick. However, the remote party must still be aware of a revocation. The revocation of a certificate will be noticed by remote parties when they attest to the VMM environment, finding the certificate revocation event in the PCR event chain. If certificates of a service partition are revoked while users are connected to it, then these connections can be torn down by the privileged partition, or – in case of a severe compromise – the privileged partition can quarantine (isolate) the service partition.

Advantages of this approach are:

- As a virtual partition is created, the TPM partition can quickly create an endpoint certificate for it.
- There is no need to purchase as many certificates through a traditional CA.
- The certificate can contain dynamic properties of the virtual partition.
- Revocation, important in a virtual environment, is greatly simplified.

5. CONCLUSIONS

We present a method of linking server endpoint validation and remote attestation using TCG quoting to avoid relay attacks, where a compromised server might relay a remote attestation quote from a trusted server.

We also construct a Platform Property certificate linking the AIK to the platform endpoint properties. This allows practical and scalable endpoint certificate revocation and rapid creation of endpoint certificates with application dependent security properties and lifetimes.

The Platform Property certificate is especially valuable in a virtualized environment, where server domains are frequently created and destroyed. It allows self signed endpoint certificates, enabling rapid and cost effective creation and scalable revocation.

6. ACKNOWLEDGMENTS

We thank Trent Jaeger, Stefan Berger, Enriquillo Valdez, and Bryan D. Payne for fruitful discussions and valuable suggestions.

7. REFERENCES

[1] T. Dierks, E. Rescorla: The Transport Layer Security (TLS) Protocol Version 1.1. April 2006.

[2] S. Kent, K. Seo: Security Architecture for the Internet Protocol. December 2005.

[3] Peter Gutmann: PKI – It's Not Dead, Just Resting. IEEE Computer Magazine, August 2002 (Vol. 35, No. 8), pp. 41-49.

[4] Trusted Computing Group. TCG TPM Specification Version 1.2. Parts I-III, 2005.

[5] Reiner Sailer, Xiaolan Zhang, Trent Jaeger, Leendert van Doorn: Design and Implementation of a TCG-based Integrity Measurement Architecture. 13th Usenix Security Symposium, San Diego, California, 2004.

[6] Trusted Computing Group. Trusted Network Connect (TNC) Architecture, Version 1.1, May 2006.

[7] Ahmad-Reza Sadeghi, Christian Stueble: Property-based Attestation for Computing Platforms: Caring about properties, not mechanisms; New Security Paradigm Workshop, 2004.

[8] Stefan Berger, Ramón Cáceres, Kenneth Goldman, Ronald Perez, Reiner Sailer and Leendert van Doorn: vTPM – Virtualizing the Trusted Platform Module. 15th Usenix Security Symposium, Vancouver, Canada, July 2006.

[9] Jonathan M. McCune, Stefan Berger, Ramón Cáceres, Trent Jaeger, Reiner Sailer: Shamon – A System for Distributed Mandatory Access Control. ACSAC, 2006.

[10] Reiner Sailer, Trent Jaeger, Xiaolan Zhang, Leendert van Doorn: Attestation-based Policy Enforcement for Remote Access. 11th ACM Conference on Computer and Communications Security (CCS), October, 2004.

[11] Tal Garfinkel, Mendel Rosenblum: A Virtual Machine Introspection Based Architecture for Intrusion Detection. Network and Distributed Systems Security Symposium, 2003.