[**Publication 6**] Zheng Yan and Peng Zhang, “A Trust Management System in Mobile Enterprise Networking”, *WSEAS Transactions on Communications*, Issue 5, Vol. 5, pp. 854-861, May 2006.

A Trust Management System in Mobile Enterprise Networking

ZHENG YAN[1], PENG ZHANG[2]
[1]Nokia Research Center, Itämerenkatu 11-13, 00180 Helsinki
[2]Nokia Enterprise Solution, Valimotie 21, 00380 Helsinki
FINLAND
zheng.z.yan@nokia.com, peng.p.zhang@nokia.com

Abstract: - Trust is crucial for mobile communications. However, how to manage trust in mobile enterprise networking among various mobile devices is problematic for companies using mobile enterprise solutions. This paper presents a trust management system in an enterprise's virtual private networks (VPN). The system supports confidential content management and overcomes the diversity support of security in different devices manufactured by different vendors. Thus, it enhances the trust in the mobile enterprise networking. Our discussion is based on a self-regulating trusted computing mechanism with the support of a root trust module inside the mobile devices. We illustrate how to apply this mechanism into mobile virtual private networks for a trusted mobile enterprise networking.

Key-Words: Trust management, Trusted computing, Enterprise networking, Security, Virtual private networking

1 Introduction

Trust plays a key role in the context of virtual private networking (VPN). However, providing advanced trust into VPN networks has proven to be problematic in mobile domains. This is mainly caused by two reasons.

First, current VPN networks lack a means to enable trust among mobile computing platforms from different manufactures. For example, an application can be trusted by Manufacture A's devices but may not be recognized by Manufacture B's devices. Moreover, from a VPN management point of view, it is difficult to manage the security of a large number of computing platforms. This problem is more serious in mobile security markets. Since different mobile device vendors provide different security solutions, it is difficult or impossible for mobile enterprise operators to manage the security of diverse devices in order to successfully run security-related services.

Second, none of existing VPN systems can ensure that the data or components on a remote user device can only be controlled according to the enterprise VPN operator's security requirements, especially during VPN connection and disconnection. The VPN server is unaware as to whether the user device platform can be trusted or not although user verification is successful. Especially, after the connection is established, the device could be compromised, which could open a door for attacks. Particularly, data accessed and downloaded from the VPN can be further copied and forwarded to other devices after the VPN connection has been terminated. The VPN client user could conduct illegal operations using various ways, e.g. disk copy of confidential files and sending emails to other people. Nowadays, the VPN operators depend on the loyalty of the VPN client users to address this potential security problem. In addition, a malicious application or a thief that stole the device could also try to compromise the integrity of the device.

Regarding the problems described above, no good solutions could be found in literatures. Related work did not consider the solutions of the problems described above [1-4]. For example, a trust management solution based on KeyNote for IPsec in [5] could ensure the trust during VPN connection in network-layer. A security policy transmission model was presented to solve security policy conflicts for large-scale VPN in [6]. But they could not help in solving the trust sustainability after the VPN connection and disconnection. Past work focused on securing network connection, not paying much attention to the necessity to control VPN terminal devices [7]. In addition, security or trust policy of the VPN operator should be different regarding different VPN client devices, which raises additional requirements for the trust management in the enterprise networking.

This paper presents a trust management system based on a virtual private network in order to enhance the trust in mobile enterprise networking. Our focus will be on how to support confidential content management and how to overcome the diversity support of security in different devices manufactured by different vendors. The discussion is based on a self-regulating trusted computing mechanism. We illustrate how to apply this mechanism into mobile virtual private networks.

The rest of the paper is organized as follows. Section 2 introduces the self-regulating trusted computing mechanism. Section 3 presents a feasible scheme of mobile self-regulating trusted computing platform (MSTCP). Section 4 describes a mobile VPN system based on the proposed mechanism and platform. Conclusions are given in the last section.

2 Self-Regulating Trusted Computing Mechanism

Generally, trust is the confidence of an entity on another entity based on the expectation that the other entity will perform a particular action important to the trustor [8-11]. Trust is subjective and dynamic. The level of trust considered sufficient is different for each individual in a certain situation. The trust relationship could be changed due to the influence of many factors. Thereby, we need a proper mechanism to support trust management not only on trust establishment, but also on trust sustaining.

In this section, we introduce the self-regulating trusted computing mechanism for establishing and sustaining the trust among computing platforms [12]. We first present the trust form used in the mechanism, and then the root trust module (RTM) on which the mechanism is based.

2.1 Trust form

The proposed mechanism uses the trust form: "Trustor A trusts trustee B for purpose P under trust conditions C based on root trust R". The element C is defined by A to specify the rules for self-regulating the trust for the purpose P, the conditions and methods to get signal of distrust behaviors, as well as the mechanism to restrict any changes at B that may influence the trust relationship. The root trust R is the foundation of A's trust on B and its sustaining. Since A trusts B based on R, it is rational for A to sustain its trust on B based on R controlled by the conditions specified by A. This form makes one-moment trust possible to be extended lasting for a period.

2.2 Root trust module

The proposed mechanism is based on a root trust module (RTM), which is also the basis of Trusted Computing Platform (TCP) [13]. The trusted computing platform (TCP) was proposed to improve the trust between users and their devices. The TCP technology ensures this through a set of hardware (HW) and software (SW) mechanisms for authenticated booting, platform integrity attestation and data access/operation control attached to platform specific configurations. The RTM could be an independent module embedded in the computing platform. It could also be a build-in feature in the current TCP's Trusted Platform Module.

The RTM at the trustee is most possibly a hardware-based security module. It has capability to register, protect and manage the conditions for trust sustaining and self-regulating. It can also monitor any computing platform's change including any alteration or operation on hardware, software and their configurations. The RTM is responsible for checking changes and restricting them based on the trust conditions, as well as notifying the trustor accordingly. Fig. 1 illustrates the basic structure of this module.

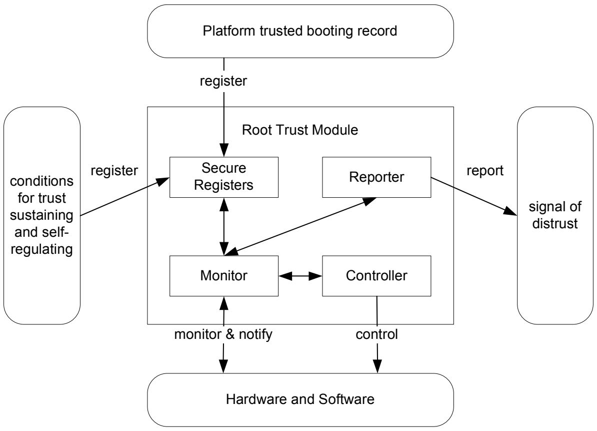


Fig. 1: Root trust module

There are two ways to know the platform changes. One is active method, that is, the platform hardware and software notify the RTM about any changes for confirmation (as described in Section 3.2). The other way is passive method, that is, the RTM monitors the changes at the hardware and software. At the booting time, the RTM registers the hash codes of each part of platform hardware and software. It also periodically calculates their run-time values and checks if they are the same as those registered. If there is any change, the RTM will check with the registered trust conditions and decide which kind of measure should be taken.

The RTM is designed and implemented inside a secure main chip of the mobile device. The secure main chip provides a Secure Environment to offer security services for operating system (OS) and software applications and some security enforcement mechanisms (e.g. system integrity booting and device identity). It also provides cryptographic functions and a secure storage. The

RTM functionalities are implemented by a number of protected applications. The protected applications are small applications dedicated to performing security critical operations inside the Secure Environment. They have strict size limitation and more resemble function libraries. The protected applications can access any resources in the Secure Environment. They can also communicate with normal applications in order to offer security services. New protected applications can be added to the system at any time. The Secure Environment software controls loading and execution of protected applications. Only signed protected applications are allowed to run.

2.3 Mechanism of self-regulating trust management

To overcome the problems presented in the introduction, we present a self-regulating trust mechanism for establishing and sustaining the trust in the computing platforms between the trustor and the trustee. The trust relationship is controlled through the conditions defined by the trustor. The conditions are ensured by the RTM at the trustee on which the trustor is willing to depend. The reasons for the trustor to depend on the RTM at the trustee can be various. Herein, we assume that the RTM at the trustee can be verified by the trustor as its expectation (through e.g. certificate verification) for some intended purpose and cannot be compromised by the trustee or other malicious entities later on. This assumption is based on the work done in industry and in academy [14-17]. Particularly, the assumption does not mean that the trustor could control the RTM at the trustee. The trust self-regulating (i.e. a trustification process) is actually conducted by the RTM based on the conditions approved by both the trustor and the trustee at the time when the trust relationship is established. In addition, the RTM is a robust module (e.g. in a silicon chip) that is protected and therefore it is hard to be compromised by any entity, including the trustee.

As shown in Fig. 2, the mechanism includes the following procedures.

- In Step 1-2, entity A challenges the RTM of entity B using a random challenging number and its certificate. The entity B verifies the entity A's certificate and responses the challenging by providing its own certificate that is the entity B's signature on the RTM's certificate and the device's certificate with the above random number. The entity A attests its correctness to ensure basic trust dependence at the entity B;
- In Step 3-6, the trust relationship is established between the entity A (trustor) and the entity B (trustee) by specifying the trust conditions and registering them at the trustee's RTM for self-regulating trust (e.g. trust sustaining);
- In Step 7-8, the trust relationship is self-regulated (e.g. sustained) between A and B through the RTM's monitor and control;
- In Step 9, the entity A could re-challenge the trust relationship if necessary when any change is against the trust conditions.

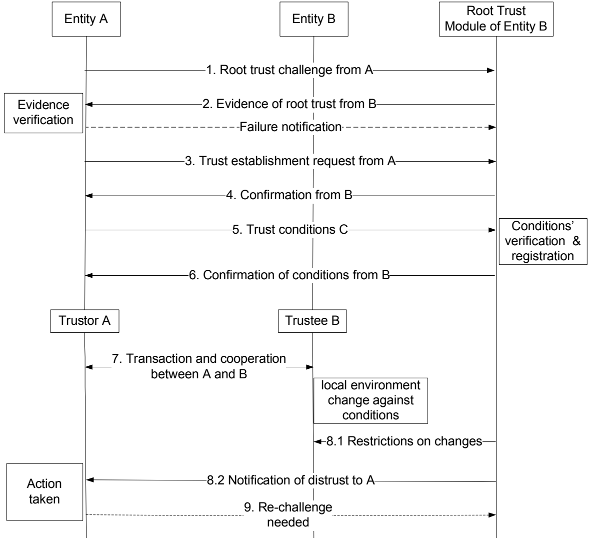

Entity A
Entity B
Root Trust Module of Entity B
1. Root trust challenge from A
Evidence verification
2. Evidence of root trust from B
Failure notification
3. Trust establishment request from A
4. Confirmation from B
5. Trust conditions C
Conditions' verification & registration
6. Confirmation of conditions from B
Trustor A
Trustee B
7. Transaction and cooperation between A and B
local environment change against conditions
8.1 Restrictions on changes
Action taken
8.2 Notification of distrust to A
9. Re-challenge needed

Fig.2: Protocol of self-regulating trusted computing mechanism

As we can see from the above protocol, the self-regulating trusted relationship is established based on the trustor's dependence on the RTM. Although the RTM is located at the trustee, its execution for trust maintenance and sustaining is based on the agreed conditions and rules approved by both the trustee and trustor at the time the trust relationship is built up. Without approval from both the trustor and the trustee, the RTM should not change any items inside the pre-registered conditions.

Notably, Step 8.2 is an option, which is applied based on the negotiation of the trust establishment. If there is no such a requirement in the trust conditions for the distrust notification, the Step 8.2 will not be applied. Otherwise, corresponding technologies or mechanisms for information protection should be further clarified by the trustor and agreed by both the trustor and the trustee in the Step 5 and 6. We could make use of public key encryption or secret key encryption to protect the notification. We could also use existing protocols to implement Step 8.2.

In order to defend against the attacks raised by information capturing and destroying, the trustee can wait for the trustor's response after the trustee sends the notification. If there is no response within an expected period, the trustee can take the corresponding measures, which are specified in the trust conditions. The corresponding measures specified in the trust conditions for any distrusted situation and any abnormal situation are trusted by the trustor.

3 A Scheme of Mobile Self-Regulating Trusted Computing Platform (MSTCP)

In this section, we further present a scheme for implementing the above mechanism targeting at a mobile device. We aim to propose a feasible solution based on the mobile device's secure main chip and operating system, thus overcome the limitations of the mobile device.

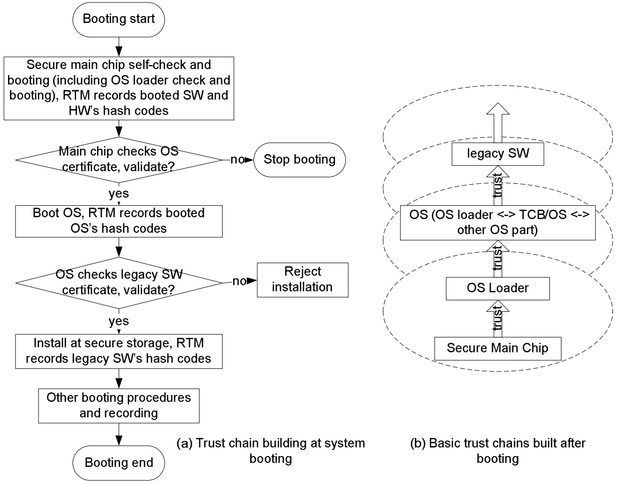
The basic idea of the scheme is to establish a computing platform internal trust management mechanism in order to support boot trust, boot trust sustaining, application trust, application trust sustaining for a up-trust chain building, and ensure down-trust if necessary through the trust assurance by the up-trust chain. The up-trust chain is a series of trust relationships built from a bottom component (e.g. a secure main chip) of a computing platform to upper components through one trust relationship establishment after another. In each trust chain, there is the chain's trustor and the trustee. The previous trust chain's trustee is actually the next trust chain's trustor. The down-trust is opposite to the up-trust in that the starting trust component is a top layer component of the computing platform or a remote computing platform. Taking platform attestation as an example (as shown in Fig. 2, step 1-6), a remote platform attests a local platform as trusted, forming down trust from the top applications, OS to the root of the up-trust chain of the local platform.

This scheme is compatible with the security mechanism of MIDP (Mobile Information Device Profile) and Symbian [18, 19], but with additional control for self-regulating trust. In particular, the RTM is deployed in the mobile device main chip with security functionalities. The establishment of a mobile self-regulating trusted computing platform includes several aspects:

a) Primary trust chain establishment during booting, (i.e. the up-trust chain's basis);
b) The up-trust chains' sustaining after booting (i.e. the down-trust chain's basis);
c) Dynamic up-trust chain establishment and sustaining at application/service level (e.g. for mobile Java applications and software components);
d) The down-trust chain support for the purpose of mobile applications and services.

In the above aspects, a) and b) solve the system level trust management, while c) and d) figure out the application level trust management.

3.1 Primary trust chain establishment during booting


Booting start
Secure main chip self-check and booting (including OS loader check and booting), RTM records booted SW and HW's hash codes
Main chip checks OS certificate, validate?
no
Stop booting
yes
Boot OS, RTM records booted OS's hash codes
OS checks legacy SW certificate, validate?
no
Reject installation
yes
Install at secure storage, RTM records legacy SW's hash codes
Other booting procedures and recording
Booting end
legacy SW
trust
OS (OS loader <-> TCB/OS <-> other OS part)
trust
OS Loader
trust
Secure Main Chip

(a) Trust chain building at system booting

(b) Basic trust chains built after booting

Fig. 3: Primary trust chain establishment

Fig. 3 (a) illustrates the trust chain establishment at the device booting. Through certificate check at the booting, the secure main chip can ensure that a trusted OS is loaded. The secure main chip offers hardware-based integrity protection features for the operating system. It can verify the integrity of the booted operating system. This kind of booting process ensures that a valid operating system image is in the memory and gains control.

After the booting, the basic trust chains could be built as shown in Fig. 3 (b). The trust chains are established based on the trust on the device main chip that contains the RTM and has a basic security support. The legacy software is applications that are installed as default by the device manufacture or used frequently as the basis for supporting other software applications. A typical example of the legacy software is a Java virtual machine (JVM).

3.2 Up-trust chain sustaining

In order to ensure the integrity and reasonable changes (e.g. normal upgrade) at the device after booting, the active method can be applied to check and ensure each trust chain's trust sustaining. The basic mechanism is that any upper layer component should report to its trustor (at lower layer) any change intention in order to get permission regarding a trusted computing platform. For example, trusted computing base (TCB), the trust kernel of the mobile device OS, should notify any change intention to the RTM at the secure main chip in order to get permission. Other parts of the device OS should report the TCB any changes for permission. The device software should report any changes to the OS for permission.

Since the software and hardware is booted as trusted, the device ensures that they have correct mechanisms for up-trust chain sustaining at the booting time. Herein, we mainly use active method to sustain the trust relationship. This is because it is more feasible for the mobile device that has limited resources. Regarding any malicious attacks that may happen after the booting, the device can monitor them through the passive method as described in Section 2.2.

3.3 Dynamic up-trust chain establishment and sustaining at application / service level

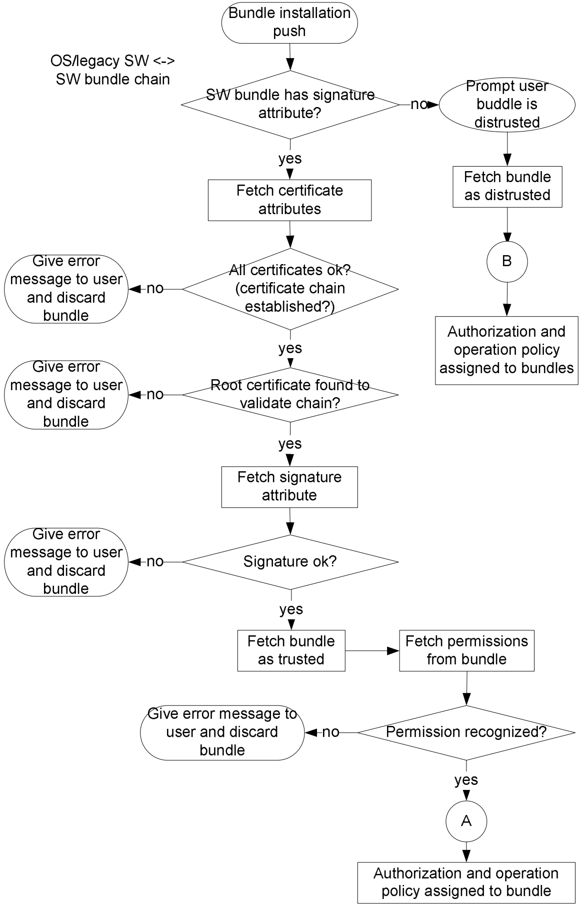

Bundle installation push
OS/legacy SW <-> SW bundle chain
SW bundle has signature attribute?
no
Prompt user buddle is distrusted
yes
Fetch certificate attributes
Fetch bundle as distrusted
B
Give error message to user and discard bundle
no
All certificates ok? (certificate chain established?)
Authorization and operation policy assigned to bundles
yes
Give error message to user and discard bundle
no
Root certificate found to validate chain?
yes
Fetch signature attribute
Give error message to user and discard bundle
no
Signature ok?
yes
Fetch bundle as trusted
Fetch permissions from bundle
Give error message to user and discard bundle
no
Permission recognized?
yes
A
Authorization and operation policy assigned to bundle

Fig. 4: Trust decision on software bundle during bundle installation

Regarding the trust chain establishment between legacy software and a software bundle, we suggest the procedure shown in Fig. 4 to make a decision on the trust of the software bundle. The software bundle is a set of software components that can be flexibly installed or uninstalled at a computing device. After identifying whether to trust or distrust the bundle, the device should authorize the bundle access permissions and assign operation policies to the bundle. This procedure is shown in Fig. 5.

The trust is also essential to be sustained in order to ensure that the installed software bundle is working in a trusted status. The software bundle trust sustaining procedure follows the same principles as in Section 3.2. Herein, the permission of change should be checked at both the legacy software and the OS in order to be consistent with their security policies. Another reason is that it is the OS (not the legacy software) that handles some application permissions and operation policies. In addition, some of the authorized permissions and the operation policies assigned to the software bundles are registered at the Legacy software. Some are registered at the OS. After the bundle is installed, a new trust chain is established, where the legacy SW trusts the SW bundles.

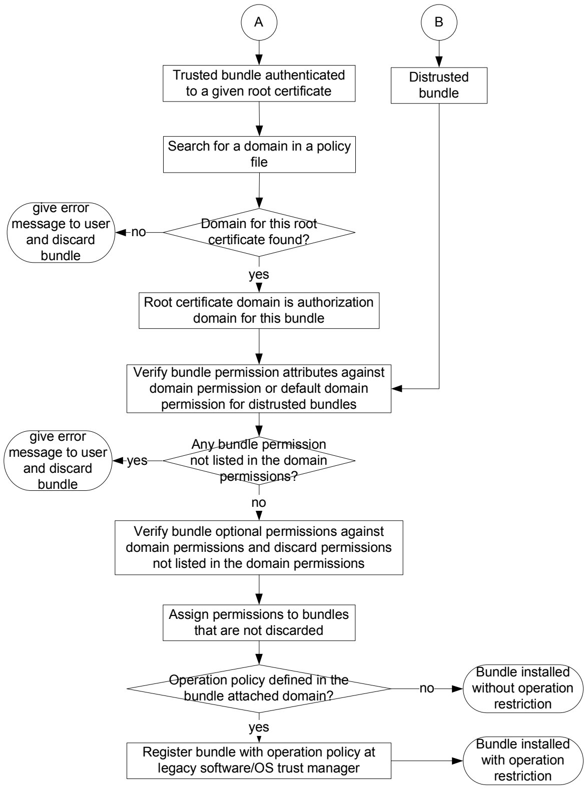

A
B
Trusted bundle authenticated to a given root certificate
Distrusted bundle
Search for a domain in a policy file
give error message to user and discard bundle
no
Domain for this root certificate found?
yes
Root certificate domain is authorization domain for this bundle
Verify bundle permission attributes against domain permission or default domain permission for distrusted bundles
give error message to user and discard bundle
yes
Any bundle permission not listed in the domain permissions?
no
Verify bundle optional permissions against domain permissions and discard permissions not listed in the domain permissions
Assign permissions to bundles that are not discarded
Operation policy defined in the bundle attached domain?
no
Bundle installed without operation restriction
yes
Register bundle with operation policy at legacy software/OS trust manager
Bundle installed with operation restriction

Fig. 5: Software bundle authorization procedure

3.4 Down trust support

For building up a trusted mobile enterprise networking, the down-trust is essential to establish between the mobile client device and the enterprise network management server. The down trust support contains two aspects: a) the trust establishment (as shown in Fig. 2); b) the embedment of the trust conditions into different trust chains at the mobile device (as shown in Fig. 6). The trust conditions will be used to check permissions for any changes that may influence the trust relationship at different up-trust chains in the mobile device.

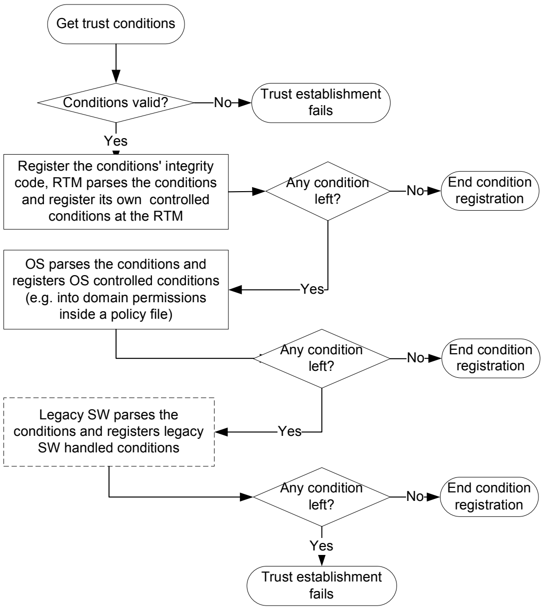

Get trust conditions
Conditions valid?
No
Trust establishment fails
Yes
Register the conditions' integrity code, RTM parses the conditions and register its own controlled conditions at the RTM
Any condition left?
No
End condition registration
Yes
OS parses the conditions and registers OS controlled conditions (e.g. into domain permissions inside a policy file)
Any condition left?
No
End condition registration
Yes
Legacy SW parses the conditions and registers legacy SW handled conditions
Any condition left?
No
End condition registration
Yes
Trust establishment fails

Fig. 6: Procedure of embedding trusted community conditions

4 Trust Management in Mobile VPN

Based on the mechanism proposed in Section 2 and the MSTCP scheme in Section 3, we provide a solution for enhancing the trust in a mobile VPN system. In this case, a VPN management server is the trustor, while a VPN client device is the trustee. A trust relationship could be established between them. The VPN management server identifies the client device and specifies the trust conditions for that type of device at the VPN connection. Thereby, the VPN client device could behave as the VPN operator expects. Additional trust conditions could be also embedded into the client device in order to control VPN-originated resources (e.g. software components or digital information originated from the VPN). Therefore, those resources could be managed later on as the VPN operator expects even if the device's connection with the VPN is terminated. Even though the VPN client device is not RTM based, the trust management server can identify it and apply corresponding trust policies in order to restrict its access to confidential information and operation.

4.1 System structure

The proposed mobile VPN system comprises a plurality of client devices, gateways and servers, part or all of which are RTM based platforms. The system provides the management of root-trust based platforms in the network, and enables verification among the platforms.

Fig. 7 illustrates the proposed mobile VPN system used in mobile networks (e.g., GSM networks). In the figure, the mobile VPN users use their mobile devices to connect to their enterprise VPN and access VPN services (e.g., emails, file sharing, etc.). The mobile devices connect to the Internet through some wireless access technology (e.g., WLAN). The VPN trust management server manages the root-trust related management issues for the mobile devices. Notably, the server may reside inside the VPN or in the Internet (protected by a firewall). The server will instruct how the mobile devices can use their RTM and for what operations. Meanwhile, the server is able to push/pull the trust conditions to the mobile devices in a secure, fast and convenient way (e.g., through SSL). With the help of the server, the mobile devices can more securely and easily set up trust relationships with other trust entities including other client devices and VPN network devices. Therefore, they are able to easily set up and maintain a trust relationship during VPN operations and even beforehand and afterwards.

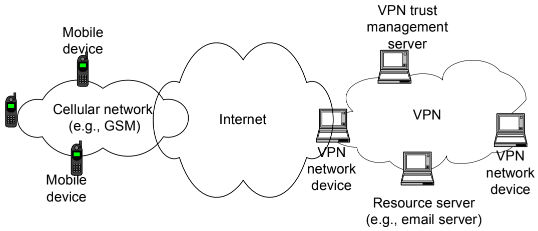

Mobile device
Cellular network (e.g., GSM)
Mobile device
Internet
VPN trust management server
VPN
VPN network device
Resource server (e.g., email server)
VPN network device

Fig. 7: Mobile VPN system structure

In particular, with the trust conditions got from the management server, the RTM with other necessary modules (e.g., secure storage) in the mobile device is able to keep and maintain the trust relationship in the device, e.g., allow or refuse to install a software, etc.

Although we only mention one trust management server, the server itself may consist of a number of servers that make the system working in practice. For example, a PKI server that generates certificates for the mobile client devices can be included into the system if needed.

4.2 System functions

The proposed system provides four major functions. Firstly, the system provides for a management server that manages the root-trust information (e.g., certificates) of various computing platforms in the network. The management server stores the root-trust information of the platforms in a local storage and is able to provide the root-trust information of any platform to other platforms upon requests. The management server also maintains the trust conditions on different platforms according to the security policy applied by the VPN operator. Those trust conditions are attached to the root-trust information of different devices and indicate the expected conditions that the device platform has to fulfill for trust establishment and management. The trust conditions can be configured at the management server in order to ensure and maintain the trust relationships with different vendor devices. In addition, the management server collects distrust notifications/warnings from the client devices and decides whether to terminate the VPN connection of the client device.

Secondly, the RTM based platform of the system is able to request the root-trust information and its trust conditions of local platforms or remote platforms from the management server. In requesting the root-trust information, the platform is also able to challenge and verify the remote platforms. By applying the trust conditions into the RTM, the challenging platform can ensure that the remote platform will work as expected as the VPN operator's specifications.

Thirdly, the RTM based platform is able to manage the trustworthiness of the platform all the time, e.g., verifying codes when the codes are installed and loaded, and verifying the RTM of remote platforms before/during communication. The platforms in the system also ensure that the VPN client device platform is the VPN operator trusted platform through the duration of the VPN connection. It restricts the distrusted change of the device hardware and software according to the VPN's connection requirements (i.e. trust conditions); therefore, a VPN trusted connection is ensured throughout the entirety of the connection.

Fourthly, with the RTM, more security related services can be provided. For example, in order to prevent crucial data (e.g. confidential files saved locally from the VPN) from being accessed in the VPN disconnection status, the usage of the data can be controlled under the RTM. This aspect is especially significant in that the employees of a company can safely use their company devices, in which company confidential data is stored, in an extranet environment (e.g., the Internet) without the potential for disclosing the crucial data to network hackers. Without this level of protection, the company devices are vulnerable to hackers via the Internet. They are also vulnerable to malicious applications and employees without loyalty.

In general, the system proposes a trust management solution in a mobile enterprise VPN context. The system aims to manage trust-related operations among devices in the network so that setting up trust across devices and between different components of a device (e.g., between applications and OS) is possible. In particular, the system ensures the execution of local platforms and remote device platforms as VPN operator's expectation by applying the trust conditions into those platforms and maintaining the trust relationship through the RTM control. Thus the system solves problems that it is hard to support multiple manufactures' devices in an enterprise VPN context. In addition, the system offers advanced control on confidential data based on the RTM after the VPN connection is terminated. Therefore, it offers enhanced trust with better security for an enterprise VPN and thus increases confidence to the users of VPN services.

4.3 Implementation

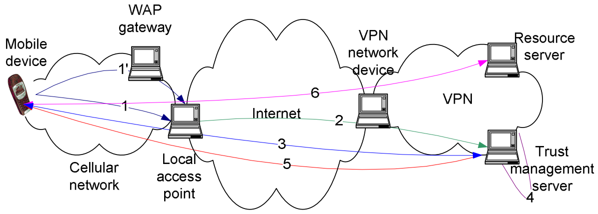


Fig. 8: An example implementation for getting trust conditions

The trust management of the proposed VPN system is driven by trust conditions issued by the management server and sent to the VPN client devices. Fig. 8 illustrates an example implementation through which a mobile device with the RTM can get the trust conditions from the management server. The conditions are embedded into the device for trust management purpose. The implementation consists of the following steps.

1. A mobile device connects (or accesses via WAP) to a local access point.

2. The local access point forwards connection request to the VPN management server. The device may also be able to connect to the VPN management server directly without passing through the access point.

3. The management server challenges the device over a secure channel (e.g., SSL) for authentication. The device may also require information from the management server for server authentication. Once the authentication succeeds, the device sends the device's information to the server upon request. The device information may include a platform

configuration certificate, and the mobile device unique platform ID.

4. The management server verifies that above documents can be trusted.

5. Then, the management server issues all kinds of files to the device. The files may include, for example, connection configurations, trust conditions for the underlying VPN connection and disconnection, trust conditions on the device for local networking (e.g. P2P enterprise networking), and trust conditions for the contents originated from the enterprise resources.

6. The device can use these files to connect to the intranet services. It also registers the conditions into the local RTM based platform for trusted VPN management (refer to Section 3).

5 Conclusions

Based on the self-regulating trusted computing mechanism and the MSTCP scheme, we illustrated how to apply them into the mobile VPN context in order to provide a trusted mobile enterprise solution. By deploying the mobile self-regulating trusted computing platform, the VPN system can be trust managed as the enterprise operator's expectation. With the proposed system, problems that retard the development of mobile enterprise networking can be solved. No matter connected or disconnected, the mobile devices behave as trusted due to the RTM control. In addition, various devices with different security solutions could work together under unified management of the trust management server.

Acknowledgments

The authors would like to thank Olli Immonen and Dr. Silke Holtmanns, Nokia Research Center, for their valuable comments.

References:

[1] Eli Herscovitz, Secure Virtual Private Networks: the Future of Data Communications, *International Journal of Network Management*, Volume 9 Issue 4, August 1999.

[2] Wood D; Stoss V; Chan-Lizardo L; Papacostas G S; Stinson M E, Virtual Private Networks, *International Conference on Private Switching Systems and Networks*, pp.132-136, Jun 1988.

[3] Regan K, Secure VPN Design Considerations, *Network Security*, pp.5-10, May 2003.

[4] Cheung K H; Misic J, On Virtual Private Network Security Design Issues, *Computer Networks*, Vol 38, No. 2, pp.165-179, 5 Feb 2002.

[5] Blaze M; Ioannidis J; Keromytis A D, Trust Management for IPsec, *ACM Transactions on Information and System Security*, Vol 5, No 2, pp.95-118, May 2002.

[6] Rongsheng Shan, Shenghong Li, Mingzheng Wang, Jianhua Li, Network Security Policy for Large-Scale VPN, *International Conference on Communication Technology Proceedings*, *ICCT* 2003, Vol 1, pp.217-220, April 2003.

[7] Hamed, H.; Al-Shaer, E.; Marrero, W., Modeling and Verification of IPSec and VPN Security Policies, *13th IEEE International Conference on Network Protocols*, pp.259 – 278, 06-09 Nov. 2005.

[8] Chopra, K.; Wallace, W.A. Trust in Electronic Environments, in *Proceedings of the 36th Hawaii International Conference on System Sciences* (HICSS'03), 2003.

[9] McKnight, D.H.; Chervany N.L. What is Trust? A Conceptual Analysis and an Interdisciplinary Model, *In Proceedings of the 2000 Americas Conference on Information Systems* (AMCI2000). AIS, Long Beach, CA, Aug 2000.

[10] McKnight, D.H.; Chervany N.L. The meanings of Trust. UMN university report. 2003. http://www.misrc.umn.edu/wpaper/wp96-04.htm

[11] Yan, Z.; Cofta, P. Methodology to Bridge Different Domains of Trust in Mobile Communications, *The First International Conference on Trust Management*, Greece, May 2003.

[12] Yan, Z.; Cofta, P. A Mechanism for Trust Sustainability among Trusted Computing Platforms, *The First International Conference on Trust and Privacy in Digital Business* (TrustBus'04), Spain, Sep 2004.

[13] TCG TPM Specification v1.2, 2003. https://www.trustedcomputinggroup.org/specs/TPM/

[14] Vaughan-Nichols, S.J. How trustworthy is trusted computing? *Computer*, Volume 36, Issue 3, March 2003.

[15] Davis P.T. TCPA: who can you trust. EDPACS: the EDP Audit, *Control and Security Newsletter*, Dec. 2002.

[16] England, P.; Lampson, B.; Manferdelli, J.; Peinado, M.; Willman, B. A Trusted Open Platform, *IEEE Computer Society*, pp.55-62, July 2003.

[17] Baldwin, A.; Shiu, S. Hardware Security Appliances for Trust. *In Proceedings of the First International Conference of Trust Management* (iTrust 2003), Crete, Greece, May 2003.

[18] MIDP 2.0 specification, Nov 2002.

[19] Dive-Reclus, C.; May, D. et al. Symbian OS Security Architecture, Nov 2003.