SECURE MOBILE DELEGATION FOR FUTURE RECONFIGURABLE TERMINALS AND APPLICATIONS

Chan Yeob Yeun (Toshiba Telecommunication Research Laboratory, Bristol, England; chan.yeun@tohsiba-trel.com); Georgios Kalogridis (Toshiba Telecommunication Research Laboratory, England; georgios.kalogridis@toshiba-trel.com);
Gary Clemo (Toshiba Telecommunication Research Laboratory, England; gary.clemo@toshiba-trel.com)

ABSTRACT

The main goal of this paper is to address applications of secure mobile delegation for future reconfigurable terminals. Additionally, a general overview will be given on past and present distributed reconfigurable mobile terminals in a Personal Area Network (PAN) context. The PAN may include a number of mobile devices which need to exchange information with each other and with their users; technologies such as Bluetooth, IrDA and WLAN could be employed. Thus, secure data transfer will be needed for properties such as confidentiality, integrity, authentication, and non-repudiation of data. However, the ability of a device to reconfigure raises a number of security issues that will need to be addressed in order to realize the potential of the reconfigurable domain. A highly distributed environment suggests the requirement for security delegation techniques. Additionally, threats increase from malicious software such as viruses, Trojan horses and worms. One can employ secure mobile delegation for securing software upgrades in reconfigurable terminals, from high level applications and system software, such as ring tones, down to lower-layer baseband modules.

1. INTRODUCTION

Many people are dependent on mobile phones, laptops, PDAs as useful tools for communicating and organizing their everyday lives. The user may also carry a number of peripherals, such as a headset or music player. Application scenarios exist in which all these devices could communicate, coming together to form a community of devices in the personal area. Technologies for realising the communication between devices could include IrDA [1], Bluetooth [2] and 802.11 (WLAN) [3].

The PAN is physically bringing computing and mobile communications together, just one factor taking us towards total convergence. The distinction between computing and communications devices will blur, with the latter adopting the flexibility that computing devices have enjoyed for many years.

Hence the concept of the Software Defined Radio; future reconfigurable devices will be dynamically reconfigurable and distributed. An important aspect of a mobile device is to support secure delegation for reconfigurable terminals in a PAN context; secure key distribution techniques are required.

Moreover, distributed mobile devices communicate using a shared broadcast medium, possibly accessible to hostile devices. In this scenario, cryptography is absolutely necessary to provide the usual security functionality including confidentiality, integrity, authentication and access control, but also secure configuration. These can be either symmetric or asymmetric cryptographic techniques or hybrids of these two as well as adopting some MExE-related work [4].

However, there are some advantages and disadvantages for each technique so one should select an appropriate technique for the secure delegation for reconfigurable terminals in PAN. Thus, we would like to consider the following security aspects:

- Security management in PAN for the terminals and application security: One should consider the distributed and dynamically reconfigurable future terminals for PAN and cover security architecture for future terminals.
- Security threats in PAN: One should also consider security properties such as confidentiality, integrity, authenticity, availability and non-repudiation.
- Comparison with Public Key Infrastructure (PKI) and symmetric key methods and with the use of authorization with certificates.

Secure delegation protocols [5-8] and reconfigurable Software Defined Radio (SDR) [9, 10] with Secure Software Download [11] concepts have been active, albeit separate, research topics for many years. The reconfiguration process will rely on obtaining requirements, capabilities and profiles from applications, devices and users, collating information from network detection or monitoring entities and downloading software components from repositories. This is potentially a highly distributed environment and the delegation of trust will be a key component to guarantee security. Existing security mechanisms do not support accountability and delegation of tasks to others. In this paper, we will address the shortcomings and introduce accountability in secure mobile delegation for reconfigurable terminals by defining new notions and proposing a new protocol that helps to eliminate the lack of accountability and trust and enhances the efficiency by reducing message passes.

2. BACKGROUND

SDR is an enabling technology applicable across a wide range of areas within the wireless industry that provides efficient and comparatively inexpensive solutions to several constraints posed in current systems. For example, SDR-enabled user devices and network equipment can be dynamically programmed in software to reconfigure their characteristics for better performance, richer feature sets, advanced new services that provide choices to the end-user and new revenue streams for the service provider. SDR is uniquely suited to address the common requirements for communications in the military, civil and commercial sectors.

2.1. Security Architecture

The Security Architecture for future terminals and applications focuses on several requirements and analysis whether these requirements can be fulfilled with PKIs and/ or PK applications. These requirements will be derived by first developing a role model for reconfigurable, distributed terminals, and then generating the requirements that each role has. This latter process will be done on the basis of common sense and experience and also by examining other collections of security requirements.

The role of PAN component administrator is not required from a service point of view, but exists to provide some sort of policing of authorization authority. We expect future terminals to consist of distributed components and to be dynamically reconfigurable. In order to describe different terminal systems we need to define a general architecture for a mobile terminal. We start by describing basic security relationships between the components in the PAN. There are two different classes of terminals:

- Smart terminals (PDA, smart phone, laptop computer, car) are expected to control and configure the PAN.
- Dumb terminals (printers, scanners, storage media, and user interface devices) provide only one function to connect the distributed terminals to smart terminals.

These two classes are expected to support a unified configuration and access control interface both at the per device level and at the PAN level. For dumb terminals this is in addition to their specialised functionality, and at a minimum it is likely to include elementary key management capability, software upgrade, and service advertisement. Depending on the configuration of the PAN and the intelligence of the terminals involved, some dumb terminals may be able to perform service discovery on their own and even request services from other devices unassisted.

2.2. Security issues in the SDR

The security aspects in SDR software downloading as well as in a number of data transactions include the following properties:

- Confidentiality ensures no information is disclosed to unauthorized entities. Messaging traveling between PAN nodes must be protected – implies using encryption techniques.
- Integrity ensures that information is never corrupted between PAN nodes – implies using cryptographic techniques for message integrity and authentication.
- Authentication ensures the identities of communication nodes in the PAN – implies using simple password techniques with biometric techniques such as fingerprint, voice recognition, retina scan etc.
- Availability means the services expected from the network or PAN nodes are available independent of denial of service attacks.
- Accountability is the property whereby the association of a principal with an object/action/right can be proved, with very high probability, to the third party.
    - Delegation of accountability can be defined as the process whereby a principal *A*, authorizes another principal *B*, to act on her behalf, by sharing a set of her rights with *B*, possibly for a specific period of time.
- Non-repudiation ensures that the origin of a message cannot deny having sent the message – implies using of digital signature techniques and appropriate protocols.

There are two main security issues for downloaded software:

- To protect the origin and integrity of the software against any accidental or deliberate corruption, and
- To provide an authorization system which enables the SDR to make an automatic decision as to whether or not to accept the downloaded software (i.e. use it to reconfigure the SDR).

Both these issues can, and are likely to be, addressed through the use of public key cryptography and PKIs. We examine how this might be achieved, and what the main issues are, in the next section.

2.3. Public Key Infrastructure (PKI) issues

PKI is to ensure the secure signaling and reconfiguration software exchanges between the parties involved. The format for public key certificates and attribute certificates, as required by the PKI, is based on the use of X.509 [12].

The addition of a digital signature to a piece of code can be used by the code's recipient to verify its correctness and origin. The public key necessary to verify the certificate can be obtained from a public key certificate, either sent with signed code or retrieved from a repository by the code's recipient. Two main issues remain:

- If the code is signed by a Certificate Authority (CA) for which the SDR does not posses the necessary public key, then a certification path (or other mechanism) will need to be deployed to enable the SDR to obtain a verified copy of the public key necessary to verify the code. Whilst mechanisms to achieve this exist, it is not clear how appropriate these will be to the mobile environment.
- Once the code has been verified, the SDR must decide whether to accept the code on the basis of the following:
    - The identity of the CA which signed the code.
    - The policy identifiers in the certificates which were verified in order to obtain the code signer's public key.
    - The policy statement built into the device by the manufacturer, together with any policy statements input by the mobile device's owner and/or user.
    - Any information associated directly with the code, e.g. that is within the scope of the signature sent with the code (this might include details of the intended scope of use of the code).

It is not clear whether policies exist which would enable complex decisions of this kind to be made automatically in a sensible way.

Finally, it is not clear whether a network operator will have any means of affecting the policies used by a mobile device to determine whether downloaded code is acceptable. If not, then this could have a damaging effect on operators if they are held responsible for damage to equipment inflicted by malicious code, and/or damage to network availability caused by devices operating in malicious ways.

3. DESIGN OF SECURE MOBILE DELEGATION FOR RECONFIGURABLE TERMINALS

The secure mobile delegation system is based on certain cryptographic techniques such as public key encryption, hashing, digital signature and symmetric encryption. We observe that currently existing secure mobile delegation systems are not designed by considering accountability in secure mobile delegation for future reconfigurable terminals. Our proposed system will address the shortcomings, introduce new notions and help to remove this lack of accountability and trust.

We assume that PKI is employed and trusted parties such as manufacturers, operators, trusted third parties and government regulators issue their certificates to mobile terminals which can store them in secure tamper resistance hardware modules, e.g. smart cards (SIM: Subscriber Identity Modules, WIM: Wireless Identity Modules, SWIM: Combined SIM and WIM, USIM: Universal Subscriber Identity Modules).

3.1. Design of a secure mobile delegation system

Existing delegation protocols [5-8] are weakened by creating unnecessary dependencies between authentication and delegation at the design phase. Even if we do not object to this in principle (there may be a specific, even if unusual, application where this is required), we want to point out that particular care must be taken in designing such protocols.

A possible shortcoming is in the use of the same key in the delegation token that user also uses to authenticate. This may look very convenient because principals can be authenticated and delegated by performing a single validation operation. However, this choice has a lot of shortcomings. It dramatically increases complexity of the key management and it weakens the robustness of the protocol.

Regarding the key management, having two different keys, one for the sole purpose of authentication and a separate one as delegation key allows the validity period of authentication and of delegation to be independent. Also, this separation allows an easier implementation of role-based models.

For example, consider cases where several different roles have been delegated to the same grantee. The grantee may need more than a single delegation key, one for each role, while she has only one single key pair for the purpose of authentication. Besides at the time at which the authentication key is bound to a principal it is usually unknown whether delegation will be required, and which rights/accountability will be delegated, thus inconsistencies may arise.

For instance, if the delegated rights/accountability have a different life time compared to the authentication key, the renewal of the key will be then very cumbersome. So it is a good practice to design authentication and delegation as separate mechanisms.

Therefore, we design our protocol in such a way that always, the grantor always chooses her delegation key pair and she never shares the signing key with other principals such as Mobile Agents.

Moreover, our novel protocols are more efficient than the previous delegation protocols in both symmetric and asymmetric techniques since our proposed protocols have less message passes than previous protocols. And our cascade delegation protocols are more compact, efficient and ideal for reconfigurable terminals than the previous protocols.

3.2. Basic Secure Delegation Protocol

In this section, we will propose the protocol which is based on tokens of delegation [5-8]. Our protocol allows Mobile Agents to delegate their own accountability to any other Mobile Agents. It assumes that each Mobile Agent possesses a priori an authentic copy of the public key pairs and access to a digital signature service. The signing key is kept secret whereas the verification key is public.

Mobile Agent *A* sends a message to Mobile Agent *B*. Mobile Agent *B* requires assurance that the message originated from Mobile Agent *A*. A diagram of the basic protocol for secure mobile delegation for future reconfigurable terminals is given in Figure 1.

Figure 1 – Basic protocol for the secure mobile delegation

The above basic protocol provides a simple solution and involves *A* sending a signed message to *B* as follows:

$$M1: A \rightarrow B: B \,\|\, T_A / N_A \,\|\, P_B(K_{P-T-E} \,\|\, \Gamma \,\|\, S_A(h(DT)))$$

Where $A \rightarrow B$, *A* sends *M1* to *B* and DT is a Delegation Token and || is concatenation of data. $P_B(Y)$ denotes the public key encryption on *Y* using *B*'s public key. S_A *(Y)* denotes the signature operation on *Y* using *A*'s signature private key and *h* denotes the one-way collision-resistance hash functions [13, 14]. The inclusion of identifier *B* in *M1* and Delegation Token ($DT = K_{P-T-E} \,\|\, B \,\|\, T_A / N_A \,\|\, \Gamma$) is necessary to prevent the token from being accepted by anyone other than the intended verifier, where $\Gamma = (R, L)$, $R =$ set of roles or tasks and $L =$ life span of the delegation token that was generated by Mobile Agent *A*. K_{P-T-E} is a Power To Execute delegation key between Mobile Agent *A* and Mobile Agent *B* can be either symmetric key or a public key that was generated by Mobile Agent *A*. The corresponding secret key that was generated by Mobile Agent *A* is kept secret. If this key is a public key then the Mobile Agent *A* has a public key used for a public encryption key and a secret key used for signing. The choice of using either a Time stamp T_A that is generated by *A* or a Nonce (Number used Once) N_A that is generated by *A* in this protocol depends on the technical capabilities of the Mobile Agents as well as on the environment.

Alternatively, if Mobile Agent *A* and Mobile Agent *B* have a pre-established relationship in the form of a shared secret k_1, a keyed-hash or Message Authentication Code (*MAC*) [15] can be that of a digital signature. In a scenario where an agent is frequently communicating with the same Mobile Agent (or Mobile Agents), this can be more efficient solution as follows:

$$M1: A \rightarrow B: B \,\|\, T_A / N_A \,\|\, E_{K_1}(K_{P-T-E} \,\|\, \Gamma \,\|\, MAC_{k_1}(DT))$$

Where, $E_{K_1}(Y)$ denotes the symmetric encryption on *Y* using the shared key K_1 between *A* and *B*. If Mobile Agent is executing on a host that is trusted and the Mobile agent's secrets (e.g. cryptographic keys resided in secure hardware modules) have not been compromised, the above protocols are enough to ensure data origin.

In order to protect against delegation token duplication and delegation token deletion, the delegation token *DT* should be constructed to include the intended recipient and a freshness value such as a timestamp, a random number and a nonce. Clock based timestamps required synchronized clocks which would be infeasible to provide securely for the platforms in our proposed protocols

3.3. Cascade Delegation Protocol

For mobile agents, the above protocols can be a basis and if there are multiple Mobile Agents, moving from its original Mobile Agent *A, to B, C,…* and finally to *Z* before returning to, or sending a final message to, the original Mobile Agent *A*.

The original Mobile Agent *A* is assumed to be fully trusted by the Mobile Agents. For example, any Mobile Agents from *A* to *Z* would be able to produce a valid signature simply by extracting the Mobile Agent's cryptographic key and using it to sign a Delegation Token, if one considers a cascade delegation that is proposed without increasing complexities and bulky message exchanges as follows:

The single stage delegation is precisely the same as the given in the case of basic protocol.

$$M1: A \rightarrow B : B \| T_A / N_A \| P_B(K_{P-T-E} \| \Gamma \| S_A(h(DT)))$$

Here is the message for a second stage delegation, from *B* to *C* as follows:

$$M2: B \rightarrow C : C \| T_B / N_B \| B \| T_A / N_A \| P_C(K_{P-T-E'} \| \Gamma' \| S_B(h(DT')) \| \Gamma \| K_{P-T-E} \| S_A(h(DT)))$$

Where $DT' = K_{P-T-E'} \| C \| T_B / N_B \| \Gamma'$ and $K_{P-T-E'}$ is a power to execute delegation key between Mobile Agent *B* and Mobile Agent *C*. $\Gamma' = (R', L')$, $R' =$ set of roles or tasks and $L' =$ life span of the delegation token that was generated by Mobile Agent *B*.

The *DT* provided by Mobile Agent *A* is nested within Mobile Agent *B*. The inclusion of identifier *C* in *M2* and *DT*' is necessary to prevent the token from being accepted by anyone other than the intended verifier as well as checking the freshness value such as Time Stamp T_B or Nonce N_B. Thus, further delegations give rise to signed *DTs* of appropriate cascading required.

Alternatively, we assume the pre-established relationship in form shared secrets k_i, *i = 1, 2,..., n* keyed-hash or *MAC*. For example, any Mobile Agent *A* to *Z* would be able to produce a valid *MAC* simply by extracting the Mobile Agent's cryptographic key and using it to *MAC* of Delegation Token, if one considers cascade delegation that is proposed as follows:

The single stage delegation is precisely the same as the given in the case of basic protocol.

$$M1: A \rightarrow B : B \| T_A / N_A \| E_{K_1}(K_{P-T-E} \| \Gamma \| MAC_{k_1}(DT))$$

Here is the message for a second stage delegation, from *B* to *C* as follows:

$$M2: B \rightarrow C : C \| T_B / N_A \| B \| T_A / N_A \| E_{K_2}(K_{P-T-E'} \| \Gamma' \| MAC_{k_2}(DT') \| \Gamma \| K_{P-T-E} \| MAC_{k_1}(DT))$$

Where $E_{K_i}(Y)$ denotes the symmetric encryption on *Y* using the shared key K_i, *i=1, 2,..., n* between *i* and *i+1*. $DT' = K_{P-T-E'} \| C \| T_B / N_B \| \Gamma'$ and $K_{P-T-E'}$ is a power to execute delegation key between Mobile Agent *B* and Mobile Agent *C*. $\Gamma' = (R', L')$, $R' =$ set of roles or tasks and $L' =$ life span of the delegation token that was generated by Mobile Agent *B*.

The *DT* provided by Mobile Agent *A* is nested within Mobile Agent *B*. The inclusion of identifier *C* in *M2* and *DT*' is necessary to prevent the token from being accepted by anyone other than the intended verifier as well as checking the freshness value such as Time Stamp T_B or Nonce N_B. Thus, further delegations give rise to signed *DTs* of appropriate cascading required.

3.4. Delegation at End Point

When an originator, Mobile Agent *A* passes rights to an intermediary and on to the last delegate, the last delegate contacts the service provider and has to prove that it holds valid *DTs* in order to request a service be granted to Mobile Agent *A*.

In order the server to verify that the service complies with these, all the *DTs* are attached. In order to trace accountability it is necessary to trace the dissemination of *DTs*. This is achieved by each party signing a particular *DT* when it passes it on to the next party and by also attaching all the signed *DTs* that have been created in a cascade delegation. The end point will be able to verify all the attached signatures but it will only be legal to only use the K_{P-T-E} as has been delegated by the last Mobile Agent in the chain. In addition, it is equally important to trace where these tokens are used. *DTs* ensure Mobile Agents with a power but not with permission to execute that power. Such permission is granted when the *DT*, following a service request, is presented to the end point and successfully checked against the access control policy in force over a secure channel such as SSL with PKI support that could provide mutual authentication, confidentiality and integrity between Mobile Agent *A* and the end points.

4. SECURITY ANALYSIS

Timestamps may be used to provide freshness and uniqueness guarantees, to detect message replay and this is necessary if security against known-key attacks is required, as this technique is otherwise vulnerable for replay attack for the unilateral key authentication protocol.

The security of timestamp-based techniques relies on use of a common time reference. This requires that host clocks be available and synchronisation is necessary to counter clock drift and must be appropriate to accommodate the acceptable time window used.

If the terminal possesses an authentic certificate for Mobile Agent *A*, the originator or operator, then the above unilateral key authentication techniques provide secure mobile delegation for future reconfigurable terminals.

In both asymmetric and symmetric cryptographic approaches, each entity maintains a key which it must keep secret except for the public key of asymmetric approach. If this key is ever compromised, then the secure delegation protocol cannot be guaranteed. So each Mobile Agent is entrusted to securely manage its own key. One of the main advantages of using the public key system is that there is no need for a trusted secret server. On the other hand by using a common symmetric key, greater performance is achieved. However, both ways will offer accountability of delegation since *DTs* are always digitally signed. Nevertheless, by using asymmetric keys then the end point can clearly confirm the origin of K_{P-T-E} .

Moreover, the attackers may be able to pretend and masquerade as a Mobile Agent if the public keys in a database are not securely protected. In the symmetric approach, the server is always trusted and should not be compromised in any way.

Our proposed protocols provide auditing mechanisms but this may not be necessary for preventing attacks rather for providing evidence subject to investigation during a possible dispute.

The delegate accountability can be granted to only Mobile Agent of the system that possesses the capabilities, that is $P_B(K_{P-T-E} \| \Gamma \| S_A(h(DT)))$ to generate such a request. If our proposed protocols suffer from the denial-of-service attack then one can avoid by relying on the natural expiration of Γ , if specified and of possibly short expiration.

5. CONCLUSION

Our novel solutions for secure mobile delegation for future reconfigurable terminals are proposed to achieve accountability in distributed networks for basic protocols, describe cascade delegation and maintain end-to-end accountability among all the involved Mobile Agents and helps to remove the lack of accountability and trust.

Moreover, in our novel solutions are more efficient than the previous solutions since our proposed protocols have less message passes than previous protocols.

Furthermore, our novel solutions are particularly useful for M-Commerce applications, where a limited amount of trust between mobile terminals in PAN environment. For example, the purchase of software components, system, or application software is to adapt the terminal's mode of operation.

Finally, we are currently working to implement our secure mobile delegation protocols in PAN scenarios by using public key based certificate as delegation tokens.

6. REFERENCES

[1] Infrared Data Association (IrDA), http://www.irda.org/

[2] Bluetooth Special Interest Group (SIG), http://www.bluetooth.com/

[3] IEEE Standard 802.11, "1999 Edition ISO/IEC 8802-5-1998, Standards for Local and Metropolitan Area Networks – Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," 1999.

[4] 3GPP TS 23.057 "Mobile Station Application Execution Environment (MExE)"

[5] M. Gasser and E. McDermott, "An architecture for practical delegation in a distributed system," In *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 20-30, 1990

[6] M. Low and B. Christianson, "Self authenticating proxies, *Computer Journal*, Vol 33, pp. 422-428, October 1994.

[7] Y. Ding, P. Horster and H. Peterson, " A new approach for delegation using hierarchical delegation token," In *Proceedings of the 2nd Conference on Computer and Communication Security*," pp 128-143, 1996.

[8] B. Crispo, "Delegation Protocols for Electronic Commerce", In the proceedings of the 6th IEEE Symposium on Computers and Communications, Hammamet,Tunisia, 3 - 5 July 2001.

[9] Software Defined Radio (SDR) Forum, http://www.sdrforum.org/

[10] "Authorization and use of Software Defined Radio: First Report and Order," U.S. Federal Communication Commission Washington, DC, September 2001.

[11] C. Yeun and T. Farnham, "Secure Software Download for Programmable Mobile User Equipment", IEE 3G Mobile Communication Technologies conference, 8-10 May 2002.

[12] ITU-T X.509 (03/00) [=ISO/IEC 9594-8:2001], "Information Technology – Open System Interconnection – The directory: Public-key and attribute certificate frameworks", 2001

[13] R. Rivest, "The MD4 message-digest algorithm," Internet Request for Comments 1320, April 1992

[14] R. Rivest, "The MD5 message-digest algorithm," Internet Request for Comments 1321, April 1992

[15] ISO 8731-1, "Banking – Approved algorithms for message authentication – Part 1:DEA", International Orgainsation for Standardization, Geneva, Switzerland, 1987