APPROACHES FOR SECURE SDR SOFTWARE DOWNLOAD

Rainer Falk (Siemens AG, Munich, Germany, rainer.falk@siemens.com)
Markus Dillinger (Siemens AG, Munich, Germany, markus.dillinger@siemens.com)

ABSTRACT

Reconfiguration involves besides setting of configuration parameters also the download of reconfiguration software. The basic approaches for secure software download are to verify that the software originated form a trusted source (signed by a trusted provider), and the execution in a controlled, restricted execution environment (sandbox). This paper describes approaches for secure download of SDR software, i.e. software that defines or modifies wireless communication properties of mobile devices.

1. INTRODUCTION

Reconfiguration allows changing properties of communication equipment that have previously been fixed by their mere design (software defined radio). The improved flexibility poses the threat that changes are made to the configuration of a device that contradict the interests and expectation of end users, network operators and service providers, equipment manufacturers and also regulatory authorities. The basic approaches for secure software download are to verify that the software originated form a trusted source (signed by a trusted provider) and the execution in a controlled, restricted execution environment (sandbox) [1]. Malicious software could otherwise invalidate essential conformance properties, and it could also lead to other types of harm. For example, it could – when suitable protection mechanisms were note utilized – circumvent other security mechanisms required for secure network access to a cellular network or a company's Intranet, it could send a user's private data to unauthorized parties, or it could call premium rate numbers in the background. Consequently, there is a strong need for a solution for the robust and secure download of reconfiguration software. Several means for secure download of radio software have been proposed in literature [2,3,4,5]. The intention of this paper is, however, targeted more on re-using available security technology for signed content that is used e.g also for secure application download. Specific to download of radio reconfiguration software are then not the security mechanisms, but the policy to be followed.

Reconfiguration is about the definition and adaptation of the configuration of reconfigurable devices to allow the optimized usage of different networks respecting their changing conditions as utilization, radio channel quality, or interference (software defined radio). The focus of this paper is on reconfigurable terminals used in a commercial environment comprising cellular networks as e.g. UMTS and public and private wireless networks (WLAN). To ensure a correct and reliable operation of the whole reconfigurable communication system, authorization of a piece of software needs to be investigated (secure download), defining which software CAN be downloaded securely. In addition, also the operational process of defining or modifying a configuration needs to be protected, see [6].

Section 2 describes signed content as a basic security mechanism for secure software download, and section 3 discusses authorization of reconfiguration software modules modifying also radio properties, also known as certification of approval. Section 5 outlines a general framework for secure download of radio reconfiguration software based on signed content.

2. SIGNED CONTENT

A well known and widely used security mechanism to protect a download software module is a digital signature. The provider of the software module attaches a digital signature to the module that can be verified by the receiving device. The digital signature ensures that the module has not been modified (integrity) and attests its provider (authentication of origin). The receiving device validates the signature of a received software module to check whether it originates from a trusted provider and to ensure that it has not been tampered with. What is specific for secure download of reconfiguration/SDR software is the policy to be followed, see section 3, defining who can provide a SDR software module that will finally be accepted and executed by a reconfigurable device.

2.1. Digital Signature

A digital signature involves a private and a public key. Only the one who has access to the private key can compute a valid signature, while everyone knowing the public key can verify the signature. Well known algorithms are RSA/PKCS#1 [7] and DSA [8]. First a digest value of

the content to be signed is computed with a cryptographic hash function as MD5 or SHA-1. Then the actual asymmetric digital signature algorithm is computed of the digest value. A widely used format for cryptographic messages as signed content is PKCS#7 resp. cryptographic message syntax (CMS) [9]. The CMS/PKCS#7 format supports inclusion of certificates needed to verify the signature, it supports several signers, and the signed content can be contained, but it has not to. So the signature and the signed content can be encoded as a single data structure, but it is as well possible that the software itself and the signature are separate (detached signature).

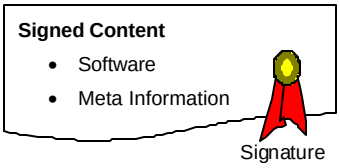

Figure 1: Signed Content

Figure 1 illustrates the concept of signed content, comprising the actual software and meta information about the software as e.g. compatibility information. Both the software and associated meta information is protected by the digital signature.

Signed content is used for example for signing MIDlet suites in MIDP2.0. What is actually signed if the Java archive (jar file). All data contained in the jar file is protected by the digital signature, including beside the actual Java code also meta information contained in the Manifest file. Instead of relying on PKCS#7, here the RSA PKCS#1 signature is encoded directly in the Java application descriptor. The encoded signature and certificates needed for signature verification are embedded in the application descriptor. This has the advantage that the application descriptor contains security information that can be verified before the actual Java archive is downloaded. To complete the verification, the actual digest of the downloaded JAR file has to be verified to match the reference digest as asserted by the digital signature.

2.2. Digital Certificate

A digital certificate binds an identifier (e.g. natural name or email address or the URL of a Web server) and a public key together. The certificate is used to verify that the used public key actually belongs to the intended person resp. Web server. A certification authority CA attests that binding between the identifier having a meaning for a human user and the public key used for cryptographic purposes by a digital signature. The client stores the public keys of root CAs that he trusts to make correct statements about the binding between identifier and public key. A certificate is accepted if it can be verified back to a trusted root CA.

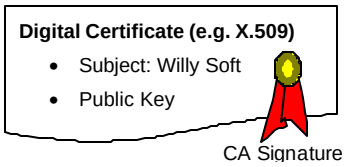

Figure 2: Example X.509 Certificate

The commonly used certificate format is X.509 [10]. Figure 2 illustrates the main elements of a digital certificate: It is basically a document containing the identifier of the subject “Willy Soft” to which the certificate is issued and the public key belonging to that subject. The digital certificate itself is signed by the certification authority CA. Further entries that are not shown indicate e.g. the validity period of the certificate or the allowed usage of the certified public key.

A certificate can be issued directly by a trusted root CA, or more generally by intermediate CAs, leading to a certificate chain. The one end of the chain is the certificate of a trusted root CA, the other end is the certificate to be verified. Starting with the certificate to be verified, the correctness of each certificate in the certificate chain is asserted by the next certificate in the chain. Using intermediate CAs allows on the one hand for flexible certification infrastructures where several intermediate CAs or peer CAs can issue certificates. On the other hand it can be used by a single CA so that its single long-lived root certificate is used only to issue more short-lived intermediate certificates that are then used to issue leaf certificates to end users.

A certificate can also be revoked, for example when a certificate has been issued by error, when the subject to which a certificate has been issued has violated the rules associated with the issued certificate or in an enterprise environment when an employee has left the company. The certification revocation status can be encoded as a certificate revocation list CRL, that is a document signed by a CA that identifies revoked certificates, or the revocation status of a single certificate can be checked online using the online certificate status protocol OCSP [11]. However, in practice revocation checks are not very common.

2.3. Authorization – Granting Permissions

A (correct) digital signature attests that the received software module is identical to the one that has been signed, i.e. that it has not been manipulated. The identifier as attested by the signer's digital certificate can be shown to a human user. This is for example done when downloading software on a PC. The end user is informed about the signer of the downloaded module and asked whether to accept the software.

For software download, also authorization information can be derived from the digital signature. The individual signer or the root certificate of the certificate chain can be used as criterion to derive whether the software module is accepted at all, and possibly the granted permissions (protection domain) when the software is executed in a controlled, managed execution environment. For example, in the MIDP2.0 recommended security policy for GSM/UMTS compliant devices, a downloaded MIDlet suite is, depending on its signer, put into one of the protection domains (manufacturer, operator, 3rd party, untrusted) [12].

3. RECONFIGURATION SOFTWARE AUTHORIZATION

While the basic mechanisms for secure software download are well known, specific for radio reconfiguration software is the policy, i.e. which party has to create the signature and thereby indicate towards a terminal that the module may be accepted (authorization, approval). The reconfigurable device receiving a radio software module has to validate the digital signature and to verify that it in facts has been computed by an entity authorized (trusted) for radio reconfiguration software of the specific category. Accessibility to the radio download security solution needs to be fixed or restricted to ensure that its properties cannot be overridden by unauthorized entity, as e.g. the end user.

3.1. Reconfiguration Classes

Specific to radio software download compared to application download is that it can define and modify properties of a radio communication interface of which regulatory authorities and network operators have an interest in its correct operation. From a regulatory perspective, it has to be ensured that essential conformance requirements are not invalidated. As the network operator wants to ensure a reliable operation and efficient and fair usage of network resources, he is concerned also about reconfiguration software that is not subject to regulatory constraints.

The main design decision is which policy is followed, depending on the reconfiguration class. It is expected that different policies are needed even within the overall range of radio-related reconfiguration. Reconfiguration software can be classified according to the stakeholder being the origin of restrictions:

- Regulatory conformance (e.g. transmission frequency, emission power)
- Network operator (e.g. monitoring and selection of most suitable radio technology, handover decisions, medium access algorithms)
- Service provider (e.g. "branding" of user interface, software needed for service-provider specific services)
- End user (e.g. applications, user interface themes, background images, ring tones)

3.2. Authorization by Device Manufacturer

As long as the downloaded software modules are specific to a single device type, e.g. a firmware update, patch or an additional feature as support for an additional radio standard, it seems to be most natural that a reconfigurable device accepts only software modules authorized by its device manufacturer (vertical market model). Here, the device manufacturer can not only ensure that conformance properties are met, but also ensure a proper operation as he is still in control on which radio software is accepted on devices he brought into the market. This approach has already been included in the MExE standard [13]. Similar to MExE, also MIDP2.0 recommended security practice for GSM/UMTS compliant devices distinguishes manufacturer, operator, third party and untrusted domains. Although MIDP 2.0 considers actually only Java MIDlets (applications), the security infrastructure (keys, certificates) could be re-used for other types of manufacturer-signed software.

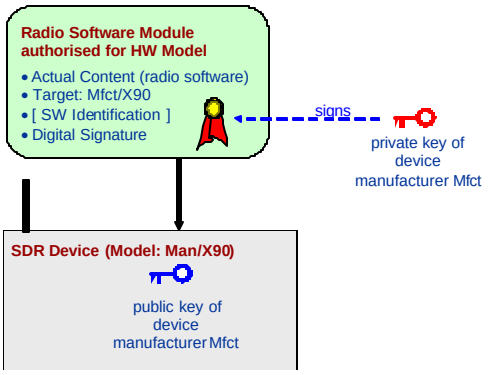

Figure 3: Software Authorization by Device Manufacturer

Figure 3 illustrates the principle of manufacturer signed software module, authorized for a specific target device. For simplicity, no certificates are used; the software module is digitally signed using directly the private key of the device manufacturer. The target device is indicated as part of the meta information, here as combination of manufacturer name (Mfct) and model (X90). But in general, other information as e.g. an approval number could be used to identify the target device for which the software module is authorized. It could e.g. also be implicitly encoded by using a different manufacturer signing key for each device model. Optionally, also a data element can be included to uniquely identify each software module, but even without an explicit identification, each software module can be identified uniquely by its cryptographic digest computed with a one-way hash function as e.g. SHA-1.

The receiving device verifies the digital signature, i.e. it ensures that the received software module has been approved (authorized) by its manufacturer and that it has not been manipulated. It also compares whether the software module has indeed been targeted for a device of its type. For this purpose, the device compares the indicated target (Mfct/X90) with its own reference identifier. The digital signature could be attached to the download module or it could be a separate, detached signature. The latter case can be advantageous when a large software module is authorized for several target devices: Download servers have to store the software module only once, and an additional small detached signature file for each target device.

3.3. Authorization of HW/SW Combination

It is interesting to note that the well-known solution for secure software download based on signed content as described above is sufficient even to realize a vertical market model where each hardware-software-combination requires authorization from an approval body, a model underlying e.g. the "Tally" download system proposed by [3]. When using the well-established signed content approach, the digital signature would be computed by a trusted approval authority, e.g. a regulatory body, instead of the device manufacturer. The reconfigurable device would have to store the approval authority's public key to be able to verify the signature, and to compare the target identifier with its own identifier.

When a radio software provider would like to get approval to use his software on a hardware device of a specific type, he would request authorization from the approval authority for this hardware-software-combination. When granted, the approval authority would compute a digital signature of the software module, including as target the intended hardware model. Optionally, also an approval number could be added as part of the meta information.

Already existing, older radio software could be authorized for a new hardware model in the same way by computing a corresponding digital signature. The hardware manufacturer, the software manufacturer, or also an independent party as a service provider or network operator could apply for approval of a hardware-software combination.

3.4. Independent Authorization of Radio HW and SW

If this approach of independent approval of radio hardware and radio software should be deemed acceptable, it could be realized using the same security technology of signed content as described above. The only required change to the software approval would be that the target meta information of a signed software module is used in a different way: It indicates not anymore a single target device. Instead, an identifier of the intended target radio execution environment would be used. The reconfigurable device would compare this identifier with the reference identifier of the implemented open radio execution environment. However, other possibilities would be to either omit the target field completely, or to use a wildcard expression matching all intended target models.

3.5. Combined Approach

The two extremes of independent authorization of radio hardware and radio software on the one side and authorization of each hardware-software-combination can be combined in a two-step solution:

- Software is approved for an open radio platform
- In addition, a compatibility check of reduced complexity is required for each intended hardware model.

The advantage of this combined approach is that it can be used to combine possibly complex and sumptuous checks of the software module against a standardized open radio platform with efficient compatibility checks to be performed for each hardware device. The combined approach could be implemented only organizationally to make conformance checks more efficient, but it could also be mapped on a technical download solution, allowing that both steps are performed independently. In particular, they could be performed by different entities, so that e.g. only the first step would involve a trusted approval body.

Also this combined approach can be secured by using digitally signed content: Here, two authorizations would be

required instead of only one. The first authorization would be targeted at the open radio execution environment, the second one at a specific target device. In particular for the second authorization, using a detached signature that is separate from the actual radio software module is advantageous.

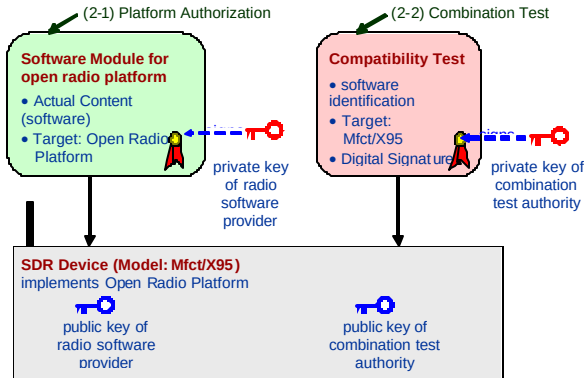

Figure 4: Software Authorization for Open Radio Platform with additional Compatibility Test

Figure 4 illustrates the combined approach: The overall authorization of a radio software module needs two separate authorizations, one asserting its authorization to be executed on the open radio platform (left hand side) and a second one (right hand side) asserting its correct operation on a specific target device. In the example shown, the first authorization is stated by the radio software provider who computes the digital signature using his private key, indicating as target the open radio platform. The second authorization is stated by an independent combination test authority, computed with its private key, indicating as target the specific target device. In the example, the first digital signature is attached to the software module, while the second one is a detached signature. It includes a unique identifier of the software, e.g. its digest or a unique name or approval number. The target device has to store the public keys of both the authorized radio software provider and of the authorized combination test authority, and it has to implement the policy to require both authorizations.

3.6. Restricted Radio Execution Environment

A controlled or managed execution environment is one where software is executed in a way where restrictions to access system functions are enforced. One objective is to isolate a downloaded radio software module from other software and from user data. Although a radio software module would modify low-level communication properties, it would not have access to other parts. More difficult is to define meaningful restrictions to enforce conformance requirements. However, as the purpose of the radio software is to define and modify low-level radio communication behavior, this comes with reduced flexibility, as certain types of reconfiguration are prevented by the controlled execution environment.

[1,5] have investigated approaches for a restricted radio execution environment. Control parameters as frequency, output power, bandwidth driving radio hardware reconfigurable by parameters can be validated to lie within an authorized range. Actual radio emissions can be monitored and compared with reference data, in particular a spectral power density mask. Reference data could be fixed or changeable only with special restrictions. These would relate to the conformance constraints that the device enforces independently of the currently executed radio software. The device itself or a communication network can monitor correct protocol behavior, e.g. obeying power control commands. The reconfiguration software of a rogue device would be terminated, and either the last correctly working software or a fixed failure mode configuration would be activated.

3.7. Activation Authorization

Even when a radio software module is authorized to be downloaded on a reconfigurable device, there may be further restrictions concerning the conditions under which it may be activated: In particular, different regulations depending on the region/location have to be respected, and possibly a radio software module requires even a dynamic authorization by the currently used network. When the same radio hardware model is marketed in different market segments (e.g. commercial wireless, public safety), the respective product should accept only the corresponding software modules to prevent for example that a commercial wireless phone accepts radio software implementing police or air traffic control communication standards.

When a single reconfigurable device shall be used in different regions (global roaming), variations in authorization rules have to be distinguished: A software module may encode as part of its meta information the regions in which it may be activated. But also the overall policy to be followed for radio software authorization may vary. For example, some regions might require approval by a regulatory authority while other regions follow a more deregulated approach. Therefore, the reconfigurable device needs to be aware of the region/location in which it is currently used, and switch to and enforce the corresponding radio software authorization policy. This policy defines in particular the trusted parties who can

authorize a software module (root certificate, public keys) and possibly restrictions based on evaluation of the software module's meta information, and possibly even the security mechanisms that may be used. When a controlled radio execution environment is used, the parameters (permissions) defining the restrictions may vary as well.

4. CONCLUSION

Existing security technology implementing security services like encryption, authentication, non-repudiation is on the one hand available and has only to be used, on the other hand developing secure security mechanisms has proven to be challenging to do right, see for example the disastrous insecurity of the WEP encryption mechanism intended to protect wireless communication. So not only for reasons of efficiency, also for reasons of security it seems to be advantageous to use existing security technology as far as possible. This affects not only the very basic cryptographic algorithms themselves, but also security standards built on top of them.

Using digital signature for secure content download is a well-known security mechanism that can also be used to protect download of radio software. The specifics of radio reconfiguration (software defined radio) do not seem to require new security mechanisms. Instead it needs to be defined *how* they shall be used to implement the policy which shall be followed:

- what needs to be signed (what to include in meta information: e.g. authorized target device(s), software identifier or approval number, region where may be used; further conditions that have to be met for activation of software module)
- who is authorized (trusted) to sign a download software module and thereby authorize/approve its use.
- required public key infrastructure
- exact format (e.g. PKCS#7 with RSA signature 1024 bit, use attached or detached signature)

It is important to notice is that the policy to be followed will vary most probably not only with local regulations, but will depend also on the evolvement of regulatory rules, the specific market for which a reconfigurable device is intended and the underlying business model. Furthermore, it will vary depending on reconfiguration classes that can be used to distinguish the properties that are to be modified resp. defined (e.g. relevant for regulatory conformance, relevant for network operation, relevant for end user). For example, while applications for a controlled execution environment might by accepted from any source under user decision, low-level radio software modules could only be accepted when approved by the device manufacturer or another trusted approval authority, without the user having a possibility to override this policy. Algorithms for cell selection and medium access could require approval by network operator to ensure a correct, fair, and efficient operation of the mobile communication system.

5. ACKNOWLEDGEMENT

This work has been performed in the framework of the EU funded project E^2R. The authors would like to acknowledge the contributions of their colleagues from E^2R consortium.

6. REFERENCES

[1] IST-SCOUT "Architecture, Functions and Security Analysis and Traffic Management Schemes for IP-Based Mobile Networks and Re-configurable Terminals in Cellular and Ad-Hoc Networks", IST-SCOUT Deliverable D4.1.2, 2004. http://www.ist-scout.org/

[2] L. Michael, M. Mihaljevic, S. Haruyama, R. Kohno: "Security Issues for Software Defined Radio – Design of a Secure Download System", *IEICE Trans.Comm*, Vol.E85-B No 12, Dec. 2002, pp. 2588–2600.

[3] Y. Suzuki, K. Oda, R. Hidaka, H. Harada, T. Hamai, T. Yokoi: "Technical Regulation Conformity Evaluation System for Software Defined Radio", *IEICE Trans. Commun.*, Vol. E86-B, No. 12, pp. 3392–3400, Dec. 2003.

[4] P. Cook: "Wireless Download Security", *SDRForum Document*, 2004-I-0069, 14 June 2004. http://www.sdrforum.org/MTGS/mtg_39_jun04/04_i_0069_v0_00_wireless_security_06_14_04.pdf

[5] K. Sakaguchi, C. Fung Lam, T. Doan, M. Togooch, J. Takada, K. Araki: "ACU and RSM Based Radio Spectrum Management for Realization of Flexible Software Defined Radio World", *IEICE Trans. Commun.*, Vol. E86-B, No. 12, pp. 3417–3424, Dec. 2003.

[6] R. Falk, M. Dillinger: "Decentralized Reconfiguration Control", Proc. of *SDR Forum Technical Conference*, Phoenix, Arizona, Nov. 2004.

[7] B. Kaliski, J. Staddon: "PKCS #1: RSA Cryptography Specifications Version 2.0", RFC2437, Oct. 1998.

[8] Digital Signature Standard (DSS), Federal Information Processing Standards (FIPS) Publication. 186, May 1994. http://www.itl.nist.gov/fipspubs/fip186.htm

[9] R. Housley: "Cryptographic Message Syntax (CMS)", RFC3369, Aug. 2002.

[10] R. Housley, W. Polk, W. Ford, D. Solo: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC3280, April 2002.

[11] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams: "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC2560, June 1999.

[12] Sun Micros.: "Mobile Information Device Profile, v2.0", JSR-118, 2002.

[13] 3GPP 23.057 "Mobile Execution Environment (MExE)", V6.2.0, 2003-9.