# OMTP

## MOBILE APPLICATION SECURITY

REQUIREMENTS FOR
MOBILE APPLICATION
SIGNING SCHEMES



| | |
|---|---|
| **VERSION:** | Version 1_3 |
| **STATUS:** | Approved for Publication |
| **DATE OF PUBLICATION** | 17th September 2007 |
| **OWNER:** | OMTP Limited |

## CONTENTS

## TABLE OF FIGURES



This document contains information that is confidential and proprietary to OMTP Limited. The information may not be used, disclosed or reproduced

without the prior written authorisation of OMTP Limited, and those so authorised may only use this information for the purpose consistent with the authorisation.

The information contained in this document represents the current view held by OMTP Limited. on the issues discussed as of the date of publication.

This document is provided “as is” with no warranties whatsoever including any warranty of merchantability, non-infringement, or fitness for any particular purpose. All liability (including liability for infringement of any property rights) relating to the use of information in this document is disclaimed. No license, express or implied, to any intellectual property rights are granted herein.

This document is distributed for informational purposes only and is subject to change without notice. Readers should not design products based solely on this document.

Each Open Mobile Terminal Platform member and participant has agreed to use reasonable endeavours to inform the Open Mobile Terminal Platform in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. The declared Essential IPR is publicly available to members and participants of the Open Mobile Terminal Platform and may be found on the “OMTP IPR Declarations” list at the OMTP members access area.

The Open Mobile Terminal Platform has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions.

Defined terms and applicable rules above are set forth in the Schedule to the Open Mobile Terminal Platform Member and Participation Annex Form.

# 1 INTRODUCTION

## 1.1 DOCUMENT PURPOSE AND SCOPE

To limit the risk of Malicious Applications (i.e. Malware) on mobile Terminals, the OMTP Security working group has defined an Application Security Framework [2] for mobile Terminals supporting an Application Execution Environment. This framework defines an execution and prompting environment for mobile Applications based on the underlying level of trust in these. Whether or not an Application is digitally signed and, if signed, by which signing authority, will indicate the associated level of trust.

In order to have an Application signed, a Developer will need to take the Application through a Signing Process, which may involve authentication of the Developer or a party on his behalf whose credentials will be linked to the certificate, agreement to legal contracts and Application testing. The Signing Scheme is seen as the entity managing such an end-to-end Signing Process.

Operators or Terminal manufacturers may either manage the Signing Process themselves or utilise third party Signing Schemes established within the industry to have Applications signed for the respective platforms.

To complement the Application Security Framework requirements defined by the OMTP security working group, this document will define a set of requirements for third party Signing Schemes to meet, such that any mobile Application signed by these schemes can be considered as Trusted. Applications of this type have a limited risk of being Malware as they been through a Signing Process that ensures that the provenance is known and traceable, and that compliance to predefined terms and conditions is assured via legal contracts.

Where operators or manufacturers deploy their own Signing Scheme for mobile Applications it is assumed that the trust in the Application Developer and their Application is derived from direct contractual relationships. Nevertheless, the requirements defined in this document shall serve as good guidance to mobile operators or manufacturers for running their own Signing Schemes.

It must be noted that for Signing Schemes to be successful they need to be accessible in terms of cost to Developers. Therefore, as with any software product, complete assurance is never possible.

Also note, the focus of this work will be solely addressing Application security related requirements and requirements necessary for widespread adoption of such Signing Schemes. Each scheme is free to deploy any services beyond that, e.g., additional Application testing to ensure all signed Applications meet minimum usability or performance requirements.

## 1.2 INTENDED AUDIENCE

This document addresses the following audiences:

All parties (in particular mobile operators, platform and Terminal vendors, Application Developers, Signing Scheme providers, certificate authorities) that have an interest in ensuring that Signing Schemes meet the minimum requirements such that Applications signed by these can be considered as Trusted by the mobile industry. As a consequence, this could result in:

- Operators requesting Applications to be signed by these schemes, thus ensuring that Applications meet certain security requirements before they are deployed
- Terminal security frameworks considering an Application's Certificate Authority (CA) as Trusted.

## 1.3 CONVENTIONS

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [1].

MUST: This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

MUST NOT: This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.

MAY: This word, or the adjective "OPTIONAL", means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

The requirements within this document are uniquely identified using the following format:

SSR.N.N-####, where:

- SSR stands for Signing Scheme Requirements
- N.N refers to the section number (“3.1”, ”3.2”, etc.).
- #### is a unique number that identifies the requirement.

.

# 2 BACKGROUND

The market share of mobile Terminals with an Application Execution Environment is steadily increasing and it can be assumed that it will continue to grow in the future.

Openness is a clear benefit for customers, Terminal manufacturers, software Developers and operators since it allows a broad range of Developers to develop rich and compelling Applications.

However, openness also presents challenges and threats, and can lead to an increase in Malware that will attempt to gain access to customers' personal data, set-up calls and unwanted data connections, or harm users or operators in other ways. Such Malware may be developed intentionally to malicious ends. However, there might be cases where the Developer had no malicious intent, with the threat being the result of a bug in the code.

To limit the risk of Malware, it is necessary to control an Application's execution on a mobile Terminal based on its level of trust. An Application is considered as Trusted where it leaves the user with a limited risk of it being Malware because it has been through a Signing Process that ensures that the provenance is known and traceable and that compliance to predefined terms and conditions is assured via legal contracts.

Signing Schemes can deliver the means to assign a level of Trust to an Application. These could be hosted by different parties (e.g. third party, operator or manufacturer).

Depending on the scheme, different levels of trust may be assigned to the signed Application.

Services provided by such schemes may include: Developer authentication, Application validation based on different requirements such as quality, usability and performance and signing of Publisher and Application Certificates.

Success of mobile Application security depends on effective mobile Terminal security policies (see OMTP Application Security Framework [2]), effective Application Signing Schemes and the adoption and support of both by stakeholders (e.g. Terminal vendors, operators and Signing Scheme providers).

The following section will present an overview of the processes and stakeholders for the Terminal-based Application signature verification and the corresponding Application Signing Process.

## 2.1 SIGNATURE VERIFICATION AND SIGNING PROCESS (INFORMATIVE)

Mobile user equipment with an Application Execution Environment allows a mobile user to run Applications other than those that have been pre-installed. Such Applications (e.g. a game) can be obtained from different sources. Users may download these from a mobile portal, execute it from an external memory or receive it from a friend via Bluetooth, etc.

Where mobile Terminals support an Application Security Framework as defined by OMTP, the operating system will validate (prior to installation and execution) which trust level can be assigned to the Application and consequently the access rights and conditions for this Application. This could mean that where an Application is not considered as Trusted, it may be restricted from access to certain key functionalities with access to others only upon user approval.

This assignment of trust is based on whether or not the Application has a valid digital signature and – where signed - the level of trust this signature is linked with. If the certificate that the Application is signed with has been revoked, installation of the Application will be denied and the user may be informed accordingly.

### 2.1.1 SIGNATURE VERIFICATION PROCESS

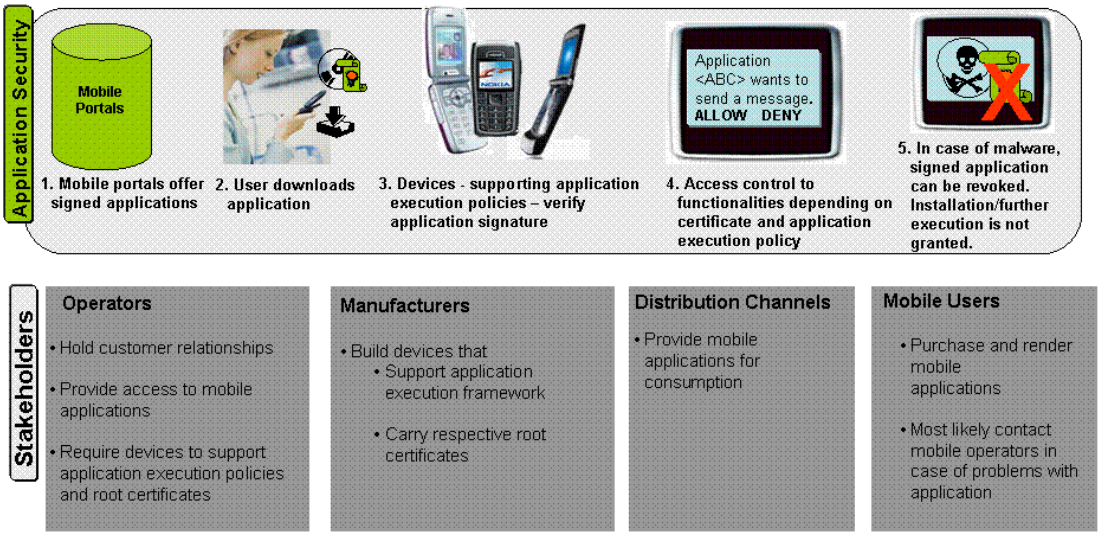


**Figure 1: Overview of Signature Verification Process.**

When a user wishes to download new Application software to her mobile Terminal, she may follow these steps (see figure 1 above). Please note, this is merely an example, there may be different experiences with different platforms. For more details on OMTP's defined security framework for mobile Terminals supporting an Application Execution Environment, see Section 7 [2]:

1. The user selects a portal that offers signed Applications. These Applications have been signed and can be downloaded to a mobile Terminal, along with the matching Application Certificates.

2. The user selects an Application to download and initiates the download of the signed Application. The Application will be delivered together with its Application Certificate and other auxiliary data such as the trust levels and the functionalities accessed by the Application.

3. The mobile Terminal determines whether the signature of the Application is valid. It may use a certificate chaining to its Trusted Root to verify a signature of the Application and auxiliary data. For example, the mobile Terminal validates the structure, content, signature and revocation status of an Application Certificate, constructs a certificate validation path to the mobile Terminal's Trusted Root, and uses the public key in this certificate to verify the integrity and authenticity of the Application software.

4. If the software has passed all these validation checks, the mobile Terminal assigns its access control privileges according to the trust level and functionality attributes encoded in the Application Certificate or auxiliary data, and the preferences set by the user, network operator and mobile Terminal manufacturer.

5. If the software fails any validation check, including the revocation status of its Application Certificate, the mobile Terminal may not grant installation of the software pending on the supported Application Security Framework.

### 2.1.2 SIGNING PROCESS

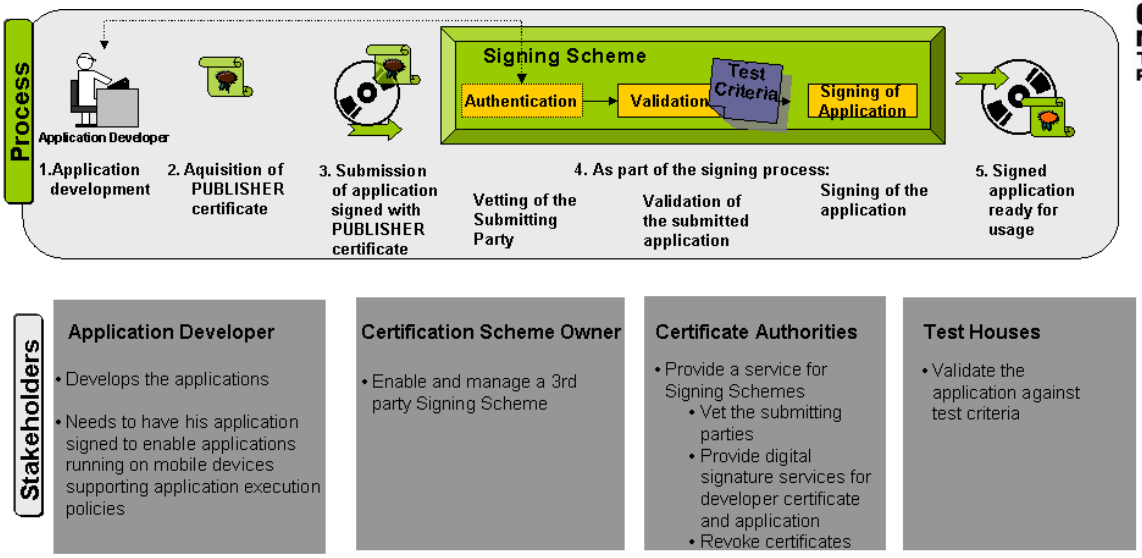


**Figure 2: Overview of Signing Process**

**The Signing Process managed by a Signing Scheme comprises of an end-to-end process enabling Developers to have their Application signed thus enabling mobile Terminals to validate the authenticity and integrity of the mobile Application.**

Where Terminals support an Application Security Framework [2] to limit risk of Malware, access to sensitive functionalities may be denied or may be granted only upon user approval unless the Application has been signed.

Hence, where an Application needs access to functionalities only granted if the Application has been signed, the Developer will need to get his Application signed so it can be validated against a Trusted Root certificate installed on the target Terminals.

Depending on the Terminal, the supporting Application Security Framework and the respective root certificate, the Developer will select the relevant Signing Scheme.

Services provided by such schemes may include: vetting and authentication of the party whose credentials are directly linked to the certificates, obtaining agreement from this party to legal contracts outlining responsibilities and liabilities, validating the Application based on different requirements such as quality, usability and integrity, signing of Publisher and Application Certificates and revocation of these if found necessary. A Signing Scheme may run the different services (and as part of this the respective identification and authentication services) either internally or outsource these to third parties. As such, a Signing Scheme may comprise the following parties:

- Publisher Certification Authority: responsible for identifying, enrolling and certifying individual Developers or agents, and managing each individual Publisher Certificate's life cycle (generate, change, revoke)

- Signing Scheme Authority: Authenticate Developer based on validating each Developer's Publisher Certificate (build the certificate chain, validate all of certificates in the chain based on fresh revocation information from Publisher Certification Authority.

- Testing house: upon request from the Signing Scheme Authority, responsible for testing the Application against predefined standard and agreed rules and criteria, and communicate the results to both Developer and Signing Scheme Authority.

- Application Certification Authority: upon request from the Signing Scheme Authority, responsible for identifying, enrolling and certifying individual mobile Applications, and managing each individual Application Certificate's life cycle (generate, change, revoke). Signing of the Publisher and Application Certificates may be done by the same or different parties.

The following provides an example of the different steps a Developer could go through (see figure 2 above).

1. The Application Developer[1] builds his Application.

2. The Application Developer (or an entity on his behalf) selects the relevant Signing Scheme and requests a Publisher Certificate. In order to obtain this, he will need to register with a Certificate Authority (CA) approved by the Signing Scheme (the Publisher CA), provide information requested to enable authentication and authorisation and enter into a legal contract outlining his responsibilities and liabilities. For the purpose of this work, this entity will be considered as the “Authorised Entity”, i.e. the party that registers with the Signing Scheme and whose credentials are directly linked to the Publisher and Application Certificate.
3. Upon successful verification, the CA will issue the Authorised Entity a Publisher Certificate containing a public key corresponding to a private key

    a. The matching private key will allow the Developer to sign his Application so that the source of origin of the Application can be linked back to the Authorised Entity.

    b. The certificate of the public key allows the Signing Scheme to verify that the Application has been signed with the Authorised Entity’s valid private key.

    c. The Publisher Certificate will capture further information such as name of the Authorised Entity and the Country/State.

[1] This could be an individual or a company.

4. The Authorised Entity signs his Application with the private key corresponding to the Publisher Certificate and submits the signed Application to the Signing Scheme. This signing does not affect the functionality of the Application but only ensures a trace back to the origin and that the Application has not been modified since. Depending on the Signing Scheme, the Application may be subject to further testing. As such, a Developer may need to select a test house, register and enter into further legal contracts in order to get his Application tested against test criteria defined by the Signing Scheme. Once the Application is approved, the Signing Scheme's Application Certification Authority will sign the Application together with auxiliary data, and issue an Application Certificate that carries Application-specific information and chains up to the root certificate associated with the intended platform. Please note that there may be other, equivalent ways of re-signing the Application. For more details, see below.

5. The signed Application and its Application Certificate and/or other auxiliary data or both may be uploaded to a portal or may be returned to the Authorised Entity for his further use. The private key used to sign the Application is not made available to the Application Developer. If it were, the Application Developer or a hacker who somehow obtained the private key could sign different software that would also appear to have been approved by the Signing Scheme.

Please note: the important characteristic of the Signing Scheme's digital signature is that it attests to the Signing Scheme's authorisation of the Application software. To verify this attestation, the Mobile Equipment (ME) will:

(a) Correctly identify the Application that the Signing Scheme has approved.
(b) Correctly identify the Trust Level and capabilities that the Signing Scheme has authorised.
(c) Authenticate the Signing Scheme's approval.

The ME accomplishes this by verifying the Signing Scheme's signature over a hash of the (a) and (b) data, and then constructing a trust chain to one of the ME's Trusted Roots (see also sections 8.1 and 8.2).

The Signing Scheme has several options for encoding the information that the ME will use to accomplish (a), (b) and (c). Two such options are discussed here.

*(i) Public Key Application Certificate*: One option is for the Signing Scheme to enclose the Application software image within a cryptographic encapsulation protocol such as S/MIME or PKCS#7, and then sign this with a private key generated specifically for this one

signing operation. Once the Application had been signed, the private key would be destroyed and the Signing Scheme would issue a public key Application Certificate that incorporates the matching public key. The Trust Level and approved capabilities information could either be incorporated into the Application Certificate as extensions, or could be included in the cryptographic encapsulation around the Application image.

*(ii) Attribute Application Certificate:* The Signing Scheme has another encoding option that also permits the ME to accomplish (a), (b) and (c). The Signing Scheme could issue the Application Certificate as an Attribute Certificate, rather than as a Public Key Certificate. In this case the Signing Scheme would place a hash of the Application software into the Application Certificate and would sign this certificate with the Signing Scheme's Attribute Authority private key. As before, the Trust Level and approved capabilities information could either be incorporated into the Application Certificate as extensions, or could be included in the encapsulation around the Application image.

Note. It is not OMTP's intention to promote either option. Both options are considered to provide an equivalent level of security and when supported by Terminals, each is seen as an acceptable solution. The requirements defined as part of this document are such that they are applicable regardless of which of the above option is used by the Signing Scheme.

Existing third party Signing Schemes include:

| SIGNING SCHEME | OPERATING SYSTEM / EXECUTION ENVIRONMENT | COMPANY RUNNING SCHEME | WEBSITE |
|---|---|---|---|
| Java Verified™ | Java J2ME MIDP | Java Verified | www.javaverified.com |
| Symbian Signed | Symbian OS | Symbian | www.symbiansigned.com |
| Mobile2Market | Pocket PC, Windows Mobile | Microsoft | http://msdn2.microsoft.com/en-gb/windowsmobile/Bb250551.aspx |
| TRUE BREW | BREW | Qualcomm | http://brew.qualcomm.com/brew/en/developer/commercialization/application_testing.html |

Signing Schemes may, however, differ in the degree of authentication, legal contracts, Application testing and revocation processes. There is no industry-agreed minimum set of requirements to which Signing Schemes must comply.

To have assurance that Applications signed by third party Signing Schemes can be Trusted in a consistent way, it is important that Signing Schemes comply with a common set of minimum requirements.

As such, the OMTP Security working group has defined a set of requirements for third party Signing Schemes to meet, such that Applications signed by these schemes can be considered as Trusted. These minimum requirements are listed in the following sections.

Note: the focus of this work will be solely addressing Application security related requirements and requirements necessary for widespread adoption. It will be up to each scheme to deploy any services beyond that, e.g., to ensure all signed Applications meet minimum usability requirements.

# 3 GENERAL REQUIREMENTS

It is important to ensure that Signing Schemes and their various stakeholders (e.g. Certification Authorities) meet the minimum requirements such that Applications signed by these provide a consistent level of trust in that the schemes provide acceptable and comparable

- Identification and Authentication of the Party submitting an Application for signing.
- Compliance with predefined terms and conditions via legal contracts.
- Revocation criteria and processes to disable the installation or (further) execution of Applications as applicable, e.g., in the case where Applications have vulnerabilities that (un)intentionally harm the user or operator. [2]
- Management of Keys and Certificate.

To achieve widespread adoption of Application signing, Developers, vendors, operators and other distribution channels (e.g. mobile content portals) need to adopt and support the schemes. As such it is important to ensure that the needs from all stakeholders are sufficiently met.

To summarise, the following elements are considered important to be supported by the Signing Scheme:

(The table below will present an overview, whilst section 4 will outline the requirements in detail).

| ELEMENT | DESCRIPTION | REQUIREMENT |
|---|---|---|
| Identification and Authentication | Validation and Authorisation of the party submitting an Application for signing to ensure traceability. | All requirements starting SSR4.1 |
| Legal Assurance | Legal contracts to cover related responsibilities and liabilities. | All requirements starting SSR4.4 |
| Application Verification | Verification of the Application to ensure no Malware is submitted. | All requirements starting SSR4.4 |

[2] It is acknowledged that unless the supporting ecosystem is built (e.g. legal processes are clarified, infrastructure and processes are set up etc.),, Signing Schemes may not be able to fully support all requirements as defined in this chapter. For more details see chapter 4.4.

| | | |
|---|---|---|
| Revocation | Support of processes and means to enable the revocation of certificates and Applications where necessary.[3] | All requirements starting SSR4.4 |
| Key Management and Certificate Processing Requirements | Requirements on the CA and Signing Scheme authority to ensure proper handling of keys and certificates | All requirements starting SSR4.5 |
| Further Requirements | Facilitate widespread adoption of Signing Scheme. | All requirements starting SSR4.6 |

[3] It is acknowledged that unless the supporting ecosystem built (e.g. legal processes are clarified, infrastructure and processes are set up etc.), Signing Schemes will not be able to fully support all requirements as defined in this chapter. For more details see chapter 4.4.

# 4 SIGNING SCHEME REQUIREMENTS

## 4.1 IDENTIFICATION AND AUTHENTICATION

### 4.1.1 BRIEF DESCRIPTION

It is important that the party submitting an Application for signing can be subsequently traced and held responsible for any damages caused by that Application. To enable this, this party will need to register for Application signing services through which it is identified and authenticated. This Authorised Entity may be an individual person or a company, and may or may not be the original Developer of the Application. The Authorised Entity may also delegate their authority to submit an Application to another entity or process if they choose, but they will remain responsible.

As such, the following definitions apply:

- The “Application Developer” is the person, persons or company that originally created the Application.
- The “Authorised Entity” is the party that registers with the Signing Scheme (or a Certificate Authority approved by the Signing Scheme) and whose credentials are directly linked to the Publisher and Application Certificate. As such, this party takes full responsibility and liability for the Applications signed.
    - Where an Application is submitted by a company, the company shall take sole responsibility for the Application with no assumption of personal liability.
    - Where an individual person has an Application they would like to get signed, this submission must be made by an Endorsing Company as the Authorised Entity. The Endorsing Company may be either his employer or another company otherwise contractually linked to him (e.g. Developer organisations). They may seek to share legal liability with the individual.[4]
- The Application Developer should be traceable in any circumstance. Where the Application Developer is not identical to the Authorised Entity, the latter takes full responsibility for the Application.
- There shall be a chain of responsibility from the Application signed via the Signing Scheme to the Authorised Entity.
- If Malware has been submitted for signing and this Malware can be linked back to the Authorised Entity, it shall be possible for any damaged party to trace the Authorised Entity for legal action.

[4] To also support the single-man Developer who may not have any formal company registration, there may be additional requirements identified in a future version.

### 4.1.2 TRACEABILITY

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.1-0010** | Any certificate (the Publisher or the Application Certificate) used within a complying Signing Scheme SHALL unambiguously identify or provide a link to the Authorised Entity. At minimum, the full name of the Authorised Entity MUST be retrievable. | |

### 4.1.3 IDENTIFICATION AND AUTHENTICATION OF THE AUTHORISED ENTITY

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.1-0020** | The Authorised Entity MUST be validated by either of the following:<br>- Confirmation of articles of incorporation *or*<br>- Registration with 3rd party databases (e.g. Dun and Bradstreet, German Handelsregister)<br>- Or equivalent[5] | |

[5]Compliance notes: Please provide the name and provider of the database used.

### 4.1.4 REGULARITY

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.1-0030** | The Publisher Certificate MUST automatically lapse after one year. After lapsing it is considered invalid for the purpose of validating any signature or for creating new signatures.<br>The expiration of the Publisher Certificate MUST NOT affect the use of Applications previously signed with the expired Publisher Certificate.<br>The Authorised Entity MAY re-apply for issuance of a new Publisher Certificate. The old public key contained in the Publisher Certificate MAY be used in the new Publisher Certificate unless it has been revoked. | |

## 4.2 LEGAL ASSURANCE

### 4.2.1 BRIEF DESCRIPTION

Across the Signing Process there can be one or several legal contracts signed with the Authorised Entity and with potentially multiple parties (e.g. Signing Scheme, CA, test house) in line with the Signing Scheme Requirements outlining responsibilities and liabilities related to:

- Information provided for authentication.
- Application submitted for signing.
- Terms and conditions that apply when the submitted Application has proven to be Malware or to contain a Security Vulnerability.

**Note: The legal contracts will be agreed between the Authorised Entity and an entity part of the Signing Scheme (e.g. Certificate Authority, test house). As such, these agreements will only cover liability in accordance with the Signing Scheme. In any other case, either other parties such as mobile users or mobile operators or other reasons outside the Signing Scheme are at stake, these issues will need to be treated outside the Signing Scheme and any reference to the Signing Scheme will not be possible**.

### 4.2.2 FORMAT

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.2-0040** | Legal contracts SHALL be documented. Examples include paper based or click through contracts. | |

Note. The following requirements listed in Section 4.2 define the principles that should be reflected in the contracts signed with the Authorised Entity as applicable. This should be reflected in future agreements and there should be no need to change already signed agreements existing at release of this document.

Furthermore it is not the intent to define exact definitions and terminology for the legal contracts. Instead it will be up to each Signing Scheme to decide on the best wording and inclusion in contracts. This will also leave each Signing Scheme with the flexibility to define any further requirements as applicable.

### 4.2.3 CORRECTNESS OF INFORMATION

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.2-0050** | Legal contracts SHALL bind the Authorised Entity to be liable for providing correct information of all requested data. | |
| **SSR.4.2-0060** | Legal contracts SHALL bind the Authorised Entity to immediately provide notification where any of the information provided for authentication related to a valid certificate changes (e.g. change of company address) until either the related certificates expire or five years after signing, whichever comes first. [6] | |

### 4.2.4 "NO MALWARE" DECLARATIONS

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.2-0070** | Legal contracts SHALL bind the Authorised Entity to declare that it has taken reasonable care to ensure that it has not introduced any Malware or a Security Vulnerability in the Application submitted for signing. | SSR.4.2-0090<br>SSR.4.2-0100 |
| **SSR.4.2-0080** | Legal contracts SHALL bind the Authorised Entity to declare that it has exercised reasonable care to ensure that no third parties have introduced any Malware or Security Vulnerabilities into the Application submitted for signing. | SSR.4.2-0090<br>SSR.4.2-0100 |

[6] It is expected that the legal contracts will state the party / the parties to whom this notification must be made.

### 4.2.5 INCLUSION OF DEFINITIONS

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.2-0090** | Legal contracts SHALL contain a definition of Malware in line with the following:<br>"Malware" means:<br>Any program code, programming instruction or set of instructions intentionally constructed with the ability to damage, interfere with or otherwise adversely affect computer programs, data files or operations, handsets, other Terminals, or the network functionalities, including, without limitation, viruses, worms, Trojan horses, spy ware, and programs deliberately carrying out a useless, disruptive, or destructive function not justified by the legitimate running of an Application, such as without limitation creating billable events (e.g. calls, SMS, network connection), changing settings, lowering security of the mobile Terminal or gathering, forwarding, manipulating, or destroying information of or about the user without appropriate permission (e.g. no permission, misleading the user to answer security related questions, etc.). | |
| **SSR.4.2-0100** | Legal contracts SHALL contain a definition of Security Vulnerability in line with the following:<br>"Security Vulnerability" means:<br>A flaw in the design or implementation of the Application which can be exploited by malicious entities to use the Application's privileges in unintended ways to damage, interfere with or otherwise adversely affect computer programs, data files or operations, handsets, other Terminals, or network functionality. | |

The requirements below in the rest of section 4.2 outline some of the actions to be taken when an Application has been found to be Malware or to have a Security Vulnerability. In order for these actions to be carried out, there must be a decision that an Application is Malware or possesses a Security Vulnerability. It is expected that the Signing Scheme itself will take this decision, and that it will use a decision process that it has specified itself,

which will include details of what constitutes proof that an Application is Malware or possesses a Security Vulnerability.

### 4.2.6 TERMS AND CONDITIONS IN CASE OF MALWARE OR SECURITY VULNERABILITY

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.2-0110** | Legal contracts SHALL bind the Authorised Entity to immediately notify the Signing Scheme or its Certification Authority or both and its distribution channel and to immediately cease, refrain and retract all use of this Application where the Application submitted for signing has proven to be Malware or has a Security Vulnerability. | SSR.4.2-0090<br>SSR.4.2-0100 |
| **SSR.4.2-0120** | Legal contracts SHALL bind the Authorised Entity to agree that where an Application submitted for signing has proven to be Malware, the Application SHALL be revoked without further consultation with the Authorised Entity. | SSR.4.2-0090 |
| **SSR.4.2-0130** | Legal contracts SHALL bind the Authorised Entity to agree that where an Application submitted for signing has proven to have a Security Vulnerability, the Application MAY be revoked without further consultation with the Authorised Entity. | SSR.4.2-0100 |
| **SSR.4.2-0140** | Legal contracts SHALL bind the Authorised Entity to agree that where an Application submitted for signing has proven to be Malware, the Authorised Entity SHALL be liable for damages caused by such Application.<br>See the Note in section 4.2.1 for further details. | SSR.4.2-0090<br>SSR.4.2-0100 |
| **SSR.4.2-0150** | Legal contracts SHALL bind the Authorised Entity to agree that in cases where an Application submitted for signing has proven to be Malware, the Publisher Certificate MAY be revoked with immediate effect. | SSR.4.2-0090 |

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.2-0160** | Legal contracts SHALL bind the Authorised Entity to agree that in cases where an Application submitted for signing has proven to have a Security Vulnerability, the Publisher Certificate MAY be revoked with immediate effect. | SSR.4.2-0100 |
| **SSR.4.2-0170** | Legal contracts SHALL bind the Authorised Entity to agree that the Signing Scheme or its CA may disclose important information (such as company, information about the Application and reason for revocation) to interested parties for their further use where an Application submitted for signing has proven to be Malware or has a Security Vulnerability. | SSR.4.2-0090<br>SSR.4.2-0100 |
| **SSR.4.2-0180** | Legal contracts SHALL bind the Authorised Entity to agree that the Signing Scheme or its CA may disclose any further relevant information to damaged third parties where an Application submitted for signing has proven to be Malware or has a Security Vulnerability. [7] | SSR.4.2-0090<br>SSR.4.2-0100 |
| **SSR.4.2-0190** | Legal contracts SHALL bind the Authorised Entity to agree that the Signing Scheme or its CA or other damaged parties MAY blacklist the Authorised Entity where an Application submitted for signing has proven to be Malware. | SSR.4.2-0090 |

[7] The damaged party will be able to claim damages only built on reasonable grounds, not just on assumptions.

### 4.2.7 TERMS AND CONDITIONS RELATED TO HANDLING OF KEYS AND CERTIFICATES

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.2-0200** | Legal contracts SHALL bind the Authorised Entity to immediately notify the Signing Scheme and/or its CA in the following cases (if they realise or if there is any suspicion that either may have realised):<br>- where the private key matching a Publisher Certificate is stored within a security Token and the Token is lost.<br>- where the private key has been cloned, made public, lost, stolen, intercepted or otherwise misdirected or disclosed. | SSR.4.4-310<br>SSR.4.4-330 |
| **SSR.4.2-0210** | Legal contracts SHALL bind the Authorised Entity to agree that his Publisher Certificate shall be revoked if the private key matching a Publisher Certificate is stored within a Token and the Token is lost. | SSR.4.4-0310 |
| **SSR.4.2-0220** | Legal contracts SHALL bind the Authorised Entity to agree that the Certificate used to sign the Application shall be revoked where its trust is based on a private key that has been cloned, made public, lost, stolen, intercepted or otherwise misdirected or disclosed.[8] | SSR.4.4-0330 |

[8] To explain: (i) In the case where a Public Key Application Certificate is used and the private key used by the Signing Scheme Authority is lost, stolen, etc., the corresponding public key Application Certificate used to sign the Application needs to be revoked. This ensures that no further Application will be signed using the compromised key; (ii) In the case where an attribute Application Certificate is used and the private key used by the Signing Scheme Authority is lost, stolen, etc., the corresponding Public Key Certificate used to sign the attribute certificate needs to be revoked. This ensures that no further Application will be signed using the compromised key.

## 4.3 APPLICATION VERIFICATION

Identification and Authentication of the Authorised Entity (section 4.1) and compliance with predefined terms and conditions ensured via legal contracts (section 4.2) are seen as the key elements providing a valuable level of assurance.

In addition, requirements verifying the authenticity and integrity of an Application (e.g. by testing for known patterns of malware) can add to the level of security.

However, it is also acknowledged that for Signing Schemes to be successful they need to be accessible in terms of cost. As such these programs are a trade off between level of assurance and costs, thus limiting the scope of Application testing.

Consequently, not all the requirements listed in this section are of a mandatory nature but are strong recommendations. It is expected that compliance with these requirements will be highly valued by potential customers of the Signing Schemes.

### 4.3.1 BRIEF DESCRIPTION

The purpose of the requirements in this section is to put high enough barriers in place so as to limit the risk that Malware is signed by the Signing Scheme or otherwise deployed.

### 4.3.2 POSITIVE DECLARATIVE STATEMENTS

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.3-0230** | When requesting signing of an Application, the Authorised Entity SHOULD declare all cases where its Application accesses any of the functional groups as defined by OMTP Application Security Framework [2] (as applicable per Application Execution Environment) and the reason for the Application accessing these. | |

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.3-0240** | When requesting signing of an Application, the Authorised Entity SHOULD provide complementing detailed information such as:<br>• Which phone numbers/URL are called, when and how often these are called.<br>• When, how often and to which addresses are messages sent.<br>• Which information is included.<br>• Which user data is read/written.<br>• Which trigger is used where an Application is automatically invoked. | SSR.4.3-0230 |
| **SSR.4.3-0250** | The Authorised Entity SHALL declare that the Application does not allow the sending of identified installation file types as part of the message.[9] | |

### *4.3.3 APPLICATION TESTING*

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.3-0260** | It SHOULD be ensured that the Application does not use any other functional group than those declared by the Authorised Entity. | SSR.4.3-0230 |

[9] It is assumed that the Signing Scheme will provide the Authorised Entity with a list outlining installation file types. This list is to be updated as applicable.

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.3-0270** | The Application SHOULD be validated against criteria, which include testing for known patterns of malicious code.<br>Where such tests are conducted, the Signing Schemes SHOULD:<br>- Ensure that testing and reporting is consistent across test houses.<br>- Ensure that the test houses shall undergo regular (e.g. annual) quality audits and blind checks.<br>- Pursue clear efforts to update test criteria as appropriate to take into account new security threats. | |

## 4.4 REVOCATION

### 4.4.1 BRIEF DESCRIPTION

It is important that Signing Schemes support processes and mechanisms to enable the revocation of the Application or Publisher Certificate or both where necessary (e.g. in the case where an Application has proven to be malicious) and to allow information sharing with interested or damaged parties as appropriate.

It needs to be ensured that revoked Applications are not installed or further distributed or that they may be disabled from (further) execution when already installed.

Whilst Certification Authorities can provide the tools to revoke, revocation requires the support of the wider ecosystem:

- Operators, Signing Schemes and Certification Authorities need to agree, establish and support an incident handling process.
- Network operators need to deploy infrastructure to support revocation (bearer, billing mechanisms).
- Terminal manufacturers need to deliver Terminals supporting revocation mechanisms.

**It is acknowledged that unless the supporting ecosystem is built, Signing Schemes will not be able to fully support all requirements as defined in this chapter**.

Activities are either ongoing or planned in GSMA and OMTP respectively to ensure progress on the various supporting elements such as Terminal support, incident handling process, etc. In line with progress of these, Signing Schemes are expected to support revocation processes and principles as defined below.

### 4.4.2 ENABLER

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.4-0280** | The Signing Scheme SHOULD enable revocation of the Publisher Certificate where necessary.[10] | |

[10] The same Publisher Certificate may be used in multiple Signing Schemes and as such agreement between the relevant schemes is necessary prior to revocation. However, in the normal case there is a direct chain between the CA and Publisher Certificate, which would avoid the issue of such other necessary coordination.

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.4-0290** | The Signing Scheme SHALL enable revocation of the Application where necessary. | |

### *4.4.3 REVOCATION CRITERIA*[11]

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.4-0300** | The Publisher Certificate MUST be revoked when requested by the Authorised Entity. | |
| **SSR.4.4-0310** | The Publisher Certificate MUST be revoked if the certificate's private key is stored on a security Token and the Token is lost. | |
| **SSR.4.4-0320** | The Publisher Certificate MAY be revoked where the Authorised Entity has been proven to submit Malware. | SSR.4.2-090 |
| **SSR.4.4-0330** | The Certificate used to sign the Application MUST be revoked where its trust is based on a private key, which has been cloned, made public, lost, stolen, intercepted or otherwise misdirected or disclosed. | SSR.4.2-220 |
| **SSR.4.4-0340** | The Application MUST be revoked where the Application has proven to be Malware. | SSR.4.2-090 |
| **SSR.4.4-0350** | The Application MAY be revoked when actively requested to do so by the Authorised Entity. | |

[11] The assumption is that revocation when found necessary shall be pursued as soon as possible.

### 4.4.4 INFORMATION PROVISIONING

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.4-0360** | When an Application has been proven to be Malware, the Signing Scheme or its CA MUST make reasonable efforts to provide at least a minimum set of information to **interested parties** (e.g. the software distribution channel) for their further use, consisting of the name of the Authorised Entity, details related to the Malicious Application and reason for revocation.[12] | SSR.4.2-090 |
| **SSR.4.4-0370** | When an Application has proven to be Malware, and upon request and in line with legal requirements, the Signing Scheme MUST make available all relevant information of the Authorised Entity and the Malicious Application to any **damaged parties**. | SSR.4.2-090 |

### 4.4.5 BLACKLISTING

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.4-0380** | In the case where either the Publisher Certificate or the Application has been revoked for reasons other than the Application being Malware (e.g the publisher certificate's private key is stored on a security token and the token is lost), the Signing Scheme MAY blacklist the Authorised Entity from any future signing services until it can demonstrate that it has taken sufficient means to correct what has led to the revocation and prevent it from happening again. | ,SSR.4.4-0300,<br>SSR.4.4-0310,<br>SSR.4.4-0320,<br>SSR.4.4-0330<br>SSR.4.4-0340<br>SSR.4.4-0350 |

[12] Note. It is up to each Signing Scheme to define further requirements as applicable to ensure compliance with local data protection laws.

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.4-0390** | In the case where either the Publisher Certificate or the Application has been revoked as a result of it being Malware, the Signing Scheme SHOULD blacklist the Authorised Entity for any future signing services until it can demonstrate that it has taken sufficient means to correct what has led to the Malware submission and prevent it from happening again. | SSR.4.2-090<br>SSR.4.4-0300,<br>SSR.4.4-0310,<br>SSR.4.4-0320,<br>SSR.4.4-0330<br>SSR.4.4-0340<br>SSR.4.4-0350 |

## 4.5 KEY MANAGEMENT AND CERTIFICATE PROCESSING REQUIREMENTS

### 4.5.1 BRIEF DESCRIPTION

It is important to ensure that the Signing Scheme and in particular its respective Certification Authorities meet a set of minimum requirements to ensure proper handling of the CA private key(s), the Publisher and Application signing keys and certificates which also requires sufficient support to the Authorised Entity, Application Developer and/or other stakeholders.

### 4.5.2 REQUIREMENTS

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.5-0400** | The Signing Scheme's Certification Authorities SHALL maintain their own certificate signing key(s) in secure, accredited hardware (FIPS 140-2 level 3 or equivalent) under at least dual control. | [3] |
| **SSR.4.5-0410** | The Signing Scheme or its Certification Authority SHALL publish a Certification Practice Statement (CPS) and Certificate Policies (CP) around its issuance of Publisher Certificates and Application Certificates, and have mechanisms in place (internal audit, external audit or accreditation) to ensure compliance with the CPS and CPs. | |
| **SSR.4.5-0420** | The Signing Scheme or its Certification Authority SHALL make a clear and meaningful statement of liability and warranty for errors contained in its certificates.[13] | |

[13] The assumption is that a statement of "No liability or warranty" as offered still qualifies as compliant.

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.5-0430** | After their usage, the Signing Scheme's Certification Authority SHALL delete the private key used to sign public key Application Certificates. The Signing Scheme SHALL take care to erase the keys such that they are non recoverable.[14] | |
| **SSR.4.5-0440** | The Signing Scheme or its Certification Authority SHALL place a requirement on the Authorised Entity to keep their Publisher Certificate private keys secret, and protect them from unauthorised usage. | |
| **SSR.4.5-0450** | The Signing Scheme or its Certification Authorities SHALL provide the Authorised Entity with adequate instructions, help and tools or utilities to achieve [SSR.4.5-0440].[15] | SSR.4.5-0440 |
| **SSR.4.5-0460** | The Signing Scheme or its Certification Authorities SHALL publish details of its signing utilities and tools and any changes thereof to ensure public confidence in its service.[16] | |

[14] If the Application Certificates are Attribute Certificates, they will not have private keys associated with them. Instead, the hash of the software would appear within a field in the Attribute Certificate.

[15] The scheme should provide signing utilities enabling the Authorised Entity to sign with a Publisher Certificate key stored in a hardware Token, if they so choose — or if they prefer, with a software key (e.g. integrated into their browser).

[16] The assumption is that such information is readily available and easily accessable to all stakeholders, (e.g. on the Signing Schemes Portal).

## 4.6 FURTHER REQUIREMENTS

### 4.6.1 BRIEF DESCRIPTION

To ensure the widespread adoption of Signing Schemes, the process needs to be efficient and Developer friendly. There needs to be cooperation between the Signing Scheme and Terminal Manufacturers and there needs to be support from the operator community.

### 4.6.2 PROCESS TRANSPARENCY

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.6-0470** | The Signing Scheme MUST have clearly defined accountabilities for:<br>• Authentication of the Authorised Entity.<br>• Holding contractual relationships.<br>• Application validation where supported.<br>• Signing of the Publisher and Application Certificate.<br>• Revocation of the Publisher Certificate and Application. | |
| **SSR.4.6-0480** | The Signing Scheme MUST make available a process overview and description that provides clear and detailed information to the Authorised Entity, Application Developers or other stakeholders for process flow, timelines, requirements including the provisions about refraining and retracting all use of Applications, contact details etc.[17] | |
| **SSR.4.6-0490** | The Signing Scheme SHALL fulfill their published service level agreements. | SSR.4.6-0470 |

[17] The assumption is that such information is readily available and easily accessible to all stakeholders, e.g. on the Signing Schemes portal.

### 4.6.3 HANDSET SUPPORT

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.6-0500** | The Signing Scheme SHOULD provide signing services for the widest possible Terminal base for Terminals using the related operating system or Application Execution Environment. | |
| **SSR.4.6-0510** | The Signing Scheme SHOULD provide sufficient support to the vendor community or operators to ensure that Terminals carry the root certificates where required.[18] | |
| **SSR.4.6-0520** | The Signing Scheme MUST provide clear and updated information on Terminals or platforms for which signing services are applicable. | |
| **SSR.4.6-0530** | Where Signing Schemes support testing of the Application, it MUST have **processes** in place so that Terminals can be available to test houses at the earliest opportunity, i.e. when Terminals are commercially available. | |
| **SSR.4.6-0540** | The Signing Scheme MUST have **processes** in place that ensure fast and efficient alignment between vendors and test houses such that test houses have latest firmware versions on the Terminals used for testing, i.e. no later than the point at which the Terminal vendors make firmware versions available publicly. | |

[18] The assumption is that it should be easy for the device vendor to have access to all information necessary to ensure support of the root certificates, have a support line and have the certificates readily available.

### 4.6.4 ALIGNMENT ACROSS SCHEMES

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.6-0550** | The Signing Scheme SHOULD pursue efforts to align authentication, validation and revocation criteria and processes across Signing Schemes as far as possible to:<br>• Enable the Authorised Entity to reuse the Publisher Certificate across schemes.<br>• Enable consistent handling of Applications being Malware or having a Security Vulnerability. | SSR.4.2-0090<br>SSR.4.2-0100 |

### 4.6.5 DEVELOPER SUPPORT

| REQ. ID | REQUIREMENT | REFERENCES |
|---|---|---|
| **SSR.4.6-0560** | The Signing Scheme SHOULD ensure that the service is acceptable to the Authorised Entity from end-to-end by taking into account price, service offering, time and customer support by monitoring customer satisfaction through regular customer surveys. | |
| **SSR.4.6-0570** | The Signing Scheme MUST provide fast and efficient customer care support.[19] | |

[19] The assumption is that customer queries should be addressed if not solved in no more than two days.

# 5 DEFINITION OF TERMS

| TERM | DESCRIPTION |
|---|---|
| **APPLICATION** | OMTP use a broad definition of "Application" in this document.<br><br>The term is used to cover active software components such as executables and .dlls as well as more passive components such as content and scripts which are included in or submitted to signing schemes. The Application may be either pre-loaded, downloaded to the mobile Terminal via the mobile network after sale, installed via another Application or transferred via infrared connection, Bluetooth, memory card or cable.<br><br>Typical examples of mobile Applications include games, media players, word processors, security Applications and content .<br><br>It excludes firmware and SIM toolkit Applications.<br><br>Depending on the Application Execution Environment, Applications may consist of one or more files with additional information such as the environment required to run the Application, debugging information, or other information used by the Application Execution Environment to prepare the program to be run. |
| **APPLICATION CERTIFICATE** | The Application Certificate is generated when the Application has met all signing requirements as defined by the Signing Scheme and is signed by the Certificate Authority or an entity on its behalf. The Application Certificate contains Application specific information and chains up to the private root CA associated with the platform, enterprise or carrier for which the Application is signed. The Application Certificate is unique to each piece of content and where applicable, must be Trusted on the end-user Terminal for secure downloading and execution. |
| **APPLICATION DEVELOPER** | The Application Developer is seen as that person, persons or company that developed the Application. |

| TERM | DESCRIPTION |
|---|---|
| **APPLICATION EXECUTION ENVIRONMENT** | The Application Execution Environment (AEE) is that layer which provides an Application with access to the functional groups and specific APIs within those functional groups. It is the AEE which restricts access to some functions for Applications of specific trust levels and provides prompts to the user as defined in the Application Security Framework [2]. |
| **ATTRIBUTE AUTHORITY** | An Attribute Authority (AA) is an entity that issues Attribute Certificates for use by other parties. It is an example of a Trusted third party. AA's are characteristic of many Privilege Management Infrastructure (PMI) schemes. |
| **ATTRIBUTE CERTIFICATE** | An Attribute Certificate is a Digital Certificate that uses a digital signature to bind information with an identity — information such as the authorisations associated with a person or a software Application, their identifying characteristics such as fingerprints, photograph, cryptographic hash, name, etc. The certificate can be used by a Privilege Verifier when making access control decisions. |
| **AUTHORISED ENTITY** | The Authorised Entity is seen as the party that authenticates with the Signing Scheme (or a CA approved by the Signing Scheme) and whose credentials are directly linked to the Publisher and Application Certificate. As such, this party takes full responsibility and liability for the Application signed with either certificate. This may be either a company or an individual authorised by a company. In the latter case, responsibility is on both parties (i.e. individual and company) to ensure full compliance with all legal terms and conditions. If necessary, both parties can be held responsible for corporate and personal liability respectively. |
| **CERTIFICATE AUTHORITY** | See Certification Authority |

| TERM | DESCRIPTION |
|---|---|
| **CERTIFICATION AUTHORITY** | A Certificate Authority or Certification Authority (CA) is an entity that issues Public Key Certificates for use by other parties. It is an example of a Trusted third party. CAs are characteristic of many Public Key Infrastructure (PKI) schemes. |
| **DEVELOPER** | See Application Developer |
| **DIGITAL CERTIFICATE** | A data object signed by a Trusted third party that signifies that the Trusted third party vouches for the accuracy of the information encoded in the data object. The term “Digital Certificate” encompasses both Public Key Certificates and Attribute Certificates. If a Digital Certificate contains a public key and also contains privilege attribute information, then the certificate is both a Public Key Certificate and an Attribute Certificate. |
| **ENDORSING COMPANY** | This company is a registered company authorising the Individual to request the signing of his Publisher or the Application Certificate. It may be either his employer or another company otherwise contractually linked to him (e.g. Developer organisations). |
| **MALICIOUS APPLICATION** | See Malware |
| **MALWARE** | Any program code, programming instruction or set of instructions intentionally constructed with the ability to damage, interfere with or otherwise adversely affect computer programs, data files or operations, handsets, other Terminals, or the network functionalities, including, without limitation, viruses, worms, Trojan horses, spy ware, and programs deliberately carrying out a useless, disruptive, or destructive function not justified by the legitimate running of an Application, such as without limitation creating billable events (e.g. calls, SMS, network connection), changing settings or gathering, forwarding, manipulating, or destroying information of or about the user without appropriate permission. This definition includes any functionality that exploits a Security Vulnerability. |

| TERM | DESCRIPTION |
|---|---|
| **PRIVILEGE MANAGEMENT INFRASTRUCTURE** | A Privilege Management Infrastructure (PMI) is similar to a Public Key Infrastructure (PKI) in that both consist of one or more Trusted third parties that issue Digital Certificates. In both cases these certificates serve as cryptographic evidence of the third party's intent to attest to a binding between one or more attributes and the subject of the certificate. The third parties within a PMI are known as Attribute Authorities and issue Attribute Certificates. |
| **PRIVILEGE VERIFIER** | A Privilege Verifier is an entity, such as a mobile platform, that relies on the information contained in an Attribute Certificate. The Privilege Verifier confirms that the Attribute Certificate is authoritative with respect to the attributes within it by constructing a valid chain of trust to one of the Privilege Verifier's Sources of Authority. The Privilege Verifier must also authenticate the certificate by constructing a valid chain of trust from the certificate to one of the Privilege Verifier's Trusted Roots. When constructing this latter chain of trust the Privilege Verifier is acting as a Relying Party. |
| **PUBLIC KEY CERTIFICATE** | A Public Key Certificate (PKC) (either encryption certificate or signature certificate) is a Digital Certificate that uses a digital signature to bind together a public key with an identity — information such as the name of a person or an organisation, their address, and so forth. The certificate can be used to verify that a public key belongs to an individual. |
| **PUBLIC KEY INFRASTRUCTURE** | A Public Key Infrastructure (PKI) is similar to a Privilege Management Infrastructure (PMI) in that both consist of one or more Trusted third parties that issue Digital Certificates. In both cases these certificates serve as cryptographic evidence of the third party's intent to attest to a binding between one or more attributes and the subject of the certificate. The third parties within a PKI are known as Certification Authorities and issue Public Key Certificates (PKCs). One of the attributes within a PKC is a public key whose matching private key is held in secret by the subject of the PKC. |

| TERM | DESCRIPTION |
|---|---|
| **PUBLISHER CERTIFICATE** | The Publisher Certificate is the Authorised Entity signing certificate and is used to sign the Application in the first instance (this will be a prerequisite prior to submitting the Application for Application signing). The Publisher Certificate is unique to each Authorised Entity and will contain related information such as the name of the Authorised Entity and the Country/State. |
| **RELYING PARTY** | A Relying Party is an entity, such as a mobile platform, that relies on the information contained in a Public Key Certificate. The Relying Party authenticates the certificate by constructing a valid chain of trust from the certificate to one of the Relying Party's Trusted Roots. |
| **SECURITY VULNERABILITY** | A flaw in the design or implementation of the Application which can be exploited by malicious entities to use the Application's privileges in unintended ways to damage, interfere with or otherwise adversely affect computer programs, data files or operations, handsets, other Terminals, or network functionality. |
| **SIGNING SCHEME** | An entity managing an end-to-end Signing Process enabling Developers to have their Application signed or thus obtaining a Digital Certificate for their mobile Application. Services provided by such schemes may include: Authentication of the party whose credentials will be linked to the certificate, agreement to legal contracts with the Authorised Entity, Application validation based on different requirements such as quality, usability and integrity, signing of Publisher and Application Certificates and revocation of these if found necessary. |
| **SIGNING PROCESS** | An end-to-end process enabling Developers to have their Application signed or thus obtaining a Digital Certificate for their mobile Application. This may involve authentication of the Party whose credentials will be linked to the certificate, agreement to legal contracts and Application testing. |

| TERM | DESCRIPTION |
|---|---|
| **SOURCE OF AUTHORITY** | A Source of Authority is an Attribute Authority identity that is implicitly trusted by a Privilege Verifier as authoritative with respect to some attribute(s). Typically one or more Sources of Authority will be installed in integrity-protected memory within a Privilege Verifier Terminal when the Terminal first subscribes to a PMI, which is often when the Terminal is manufactured. |
| **TOKEN** | A security Token (or sometimes a hardware Token, authentication Token or cryptographic Token) may be a physical Terminal that an authorised user of computer services is given to aid in authentication. The term may also refer to software Tokens. Hardware Tokens are typically small enough to be carried in a pocket or purse and often are designed to attach to the user's keychain. Some may store cryptographic keys, like a digital signature, or biometric data, like a fingerprint. Some designs feature tamper resistant packaging, others may include small keypads, thus allowing entry of a PIN. |
| **TRUSTED** | See Trusted Application |
| **TRUSTED APPLICATION** | An Application is considered as Trusted where it leaves the user with a limited risk of Malware as it has been through a Signing Process. The Signing Process ensures that the source of origin is known and traceable and compliance to predefined terms and conditions is assured via legal contracts. |
| **TRUSTED ROOT** | A Trusted Root is a Certification Authority public key that is implicitly Trusted by a Relying Party. Typically one or more Trusted Roots will be installed in integrity-protected memory within a Relying Party Terminal when the Terminal first subscribes to a PKI, which is often when the Terminal is manufactured. |

# 6 ABBREVIATIONS

| ABBREVIATION | DESCRIPTION |
|---|---|
| **API** | Application Programming Interface |
| **CA** | Certification Authority |
| **CP** | Certificate Policies |
| **CPS** | Certification Practice Statements |
| **FIPS** | Federal Information Processing Standards |
| **GSM** | Global System for Mobile communications |
| **GSMA** | GSM Association |
| **IPR** | Intellectual Property Rights |
| **ME** | Mobile Equipment |
| **OMTP** | Open Mobile Terminal Platform |
| **PKC** | Public Key Certificate |
| **PKI** | Public Key Infrastructure |
| **PMI** | Privilege Management Infrastructure |
| **S/MIME** | Secure Multipurpose Internet Mail Extensions |

# 7 REFERENCED DOCUMENTS

| NO. | DOCUMENT | AUTHOR | DATE |
|---|---|---|---|
| **1** | RFC 2119 “Key words for use in RFCs to Indicate Requirement Levels”. http://www.ietf.org/rfc/rfc2119.txt | S Bradner | March 1997 |
| **2** | OMTP Application Security Framework v2.1 (http://www.omtp.org) | OMTP | 4th September 2007 |
| **3** | FIPS 140-2. (http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf) | National Institute of Standards and Technology (NIST) | |

# 8 APPENDIX 1 – TYPES OF CERTIFICATES

Although the Signing Scheme is free to use any Digital Certificate standard, the text in this document employs the terminology used within ITU-T standard X.509. For that reason, the following taxonomy diagram may be helpful in understanding the various types of certificates discussed within this document. The gold objects are certificates whose usages are defined within this document. The blue lower case objects are certificate types defined within X.509. In X.509 terms, the Publisher Certificate is an end entity signature public key Digital Certificate (also known as a signature certificate). The Application Certificate could be of the same type, as explained in Section 8.1 below. As an alternative, the Application Certificate could be implemented as an end entity attribute Digital Certificate (also known as an Attribute Certificate), as explained in Section 8.2 below. If a Public Key Certificate is chosen to implement the Application Certificate, then the Signing Scheme Certificate must be a Certification Authority certificate. This is because, according to X.509, Public Key Certificates can be issued only by certification authorities, which are identified by means of Certification Authority certificates. If the Application Certificate is an Attribute Certificate, then the Signing Scheme Certificate must be an Attribute Authority certificate, because according to X.509 Attribute Certificates are issued by attribute authorities.

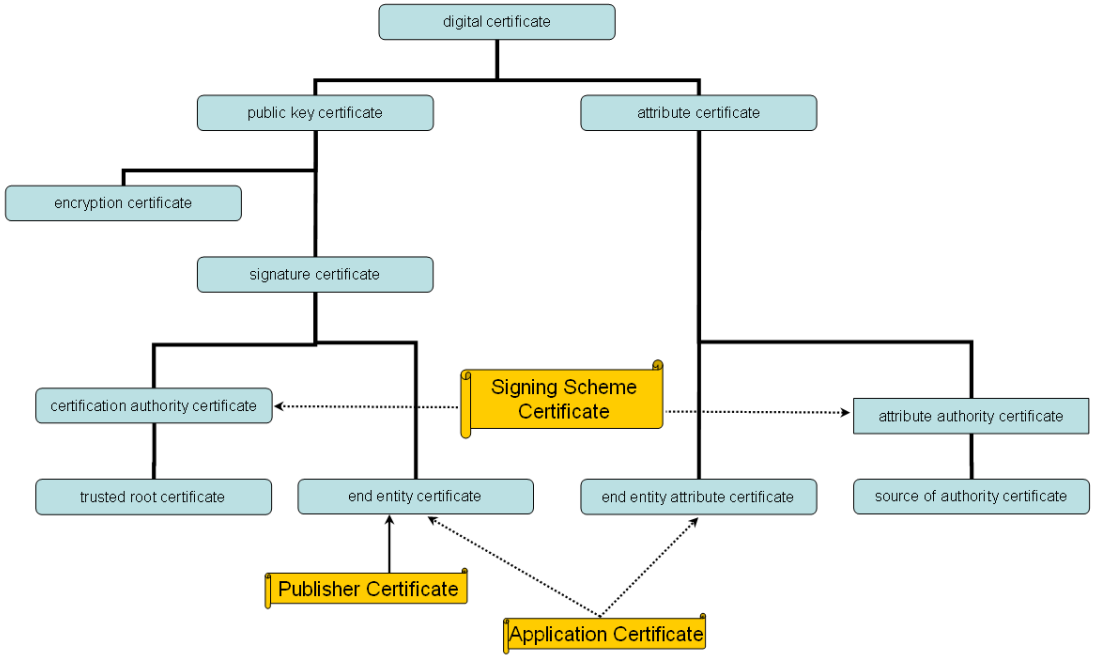


**Figure 3: Overview of Application and Attribute Certificate***

[*The *gold boxes* provide examples of the *blue boxes* ]

## 8.1 PUBLIC KEY APPLICATION CERTIFICATE

Implementing an Application Certificate as a Public Key Certificate: if the Signing Scheme is a Certification Authority and the Application Certificate is a public key signature certificate, the Application Certificate can be used both to authenticate the Application software and to verify the Signing Scheme's authorisation of the software. The software would be signed by the Signing Scheme using the private key associated with the Application Certificate. The private key must be handled very carefully because anyone who somehow obtained the private key could sign non-approved software that would also validate against the same Application Certificate. To prevent this, the key pair should be generated at the Signing Scheme and the private key should never leave the Signing Scheme's control. Since the private key is not needed to authenticate the software, once the software has been signed the private key should be carefully destroyed to prevent its compromise.

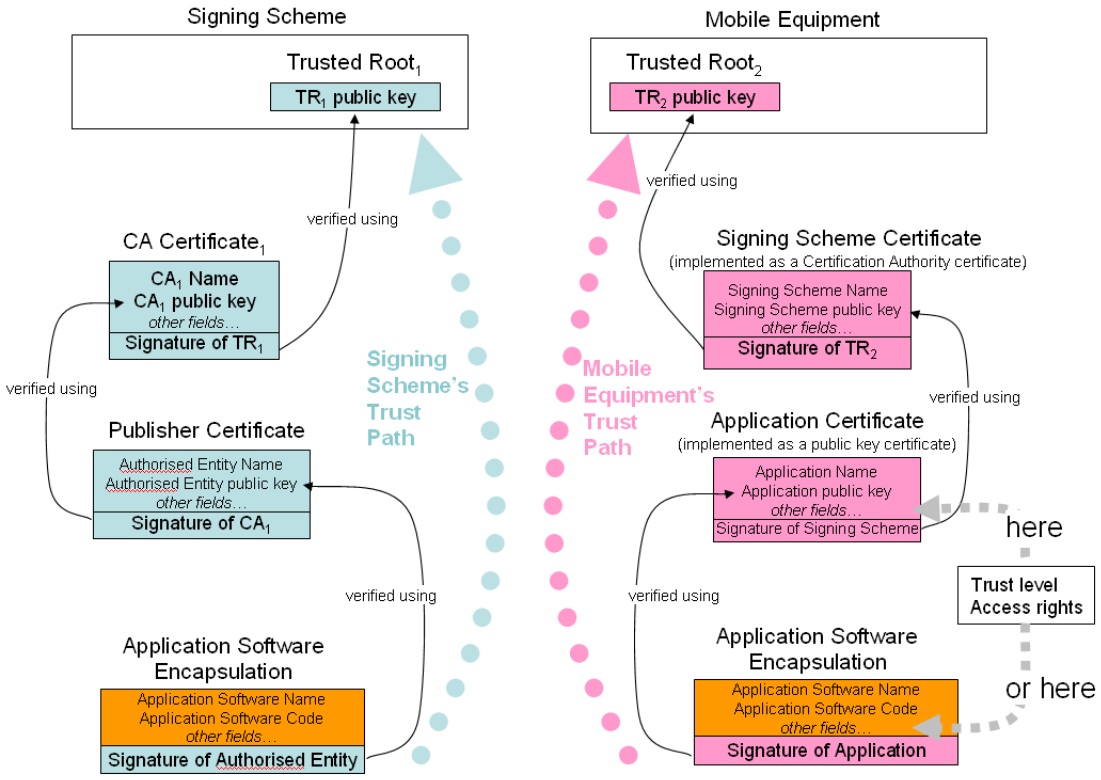


**Figure 4: Trust Path for Validating the Publisher and Application Certificate**

## 8.2 ATTRIBUTE APPLICATION CERTIFICATE

Implementing an Application Certificate as an Attribute Certificate: an Attribute Certificate is similar to a Public Key Certificate, but contains no public key. It is used to bind an attribute (in this case the hash of the software) to the subject of the certificate (the ID of the software), along with various other

information such as the Publisher ID, the trust level, and the functionalities accessed by the software. If the Signing Scheme implements the Application Certificate as an Attribute Certificate, the mobile platform would verify the Signing Scheme's authorisation of the software by means of the Application Certificate, but would authenticate the software by comparing a hash over the software with a hash value encoded in the Application Certificate. The primary advantage of using an Attribute Certificate is that the Signing Scheme does not need to be a Certification Authority within a Public Key Infrastructure (PKI). Instead, the Signing Scheme would be an Attribute Authority within a Privilege Management Infrastructure (PMI). The practical benefit of this is the Signing Scheme may have fewer legal and financial obligations than a Certification Authority would.

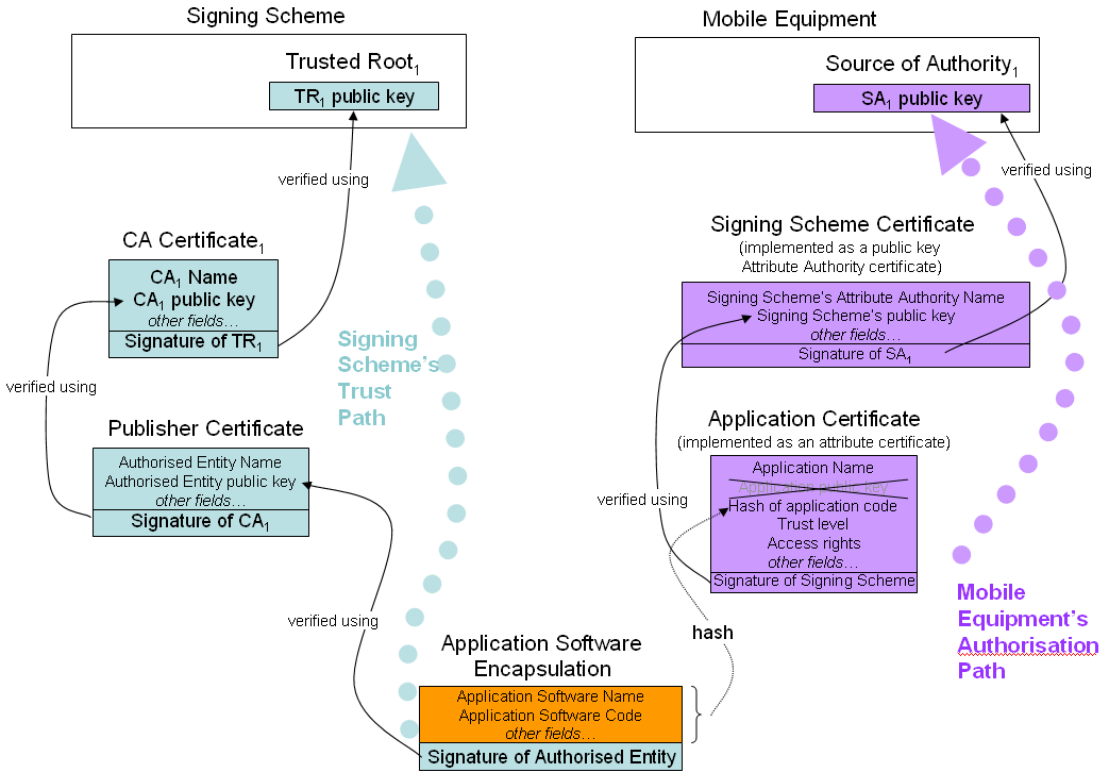


**Figure 5: Trust Path for Validating the Publisher and Attribute Certificate**

# 9 APPENDIX 2 – REQUIREMENT ID CHANGES

None of the requirements have been changed between version 1.23 (previous version) and version 1.3 (this version)

**End of Document.**