ETSI TR 133 905 V12.0.0 (2014-10)

Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
LTE;
Recommendations for Trusted Open Platforms
(3GPP TR 33.905 version 12.0.0 Release 12)

Reference
RTR/TSGS-0333905vc00

Keywords
GSM,LTE,SECURITY,UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Report (TR) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**may not**", "**need**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

Securing the storage, processing, and input and output of sensitive data on an open platform are of critical importance. Also, isolation of applications that are managing (U)SIMs and (U)SIM readers, EAP-SIM and EAP-AKA protocols, and SAP applications from untrusted applications is imperative. Protecting the interface between the trusted open platform and the UICC is also of critical importance.

Therefore, it is very much desirable that the Open Platform must have secure authentication and authorization mechanisms to protect against eavesdropping, and malicious modification of sensitive data and operator applications residing on the Open Platform.

Consequently, for the diverse 3GPP usage models of the Open Platform, such as the ones described in 3GPP TS 33.234, appropriate trust recommendations need to be outlined to counteract the threats. This document describes trust recommendations for the usage models described in 3GPP.

1 Scope

This technical report investigates relevant trust standards and technologies, both existing as well as the ones that are work-in-progress. It develops the recommendations for trusted open platforms for delivery of new applications and services to open platforms.

2 References

The following documents contain provisions, which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document.*

[1] 3GPP TS 33.220: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture".

[2] 3GPP TS 33.234: "3rd Generation Partnership Project; Technical Specification Group Service and System Aspects; 3G Security; Wireless Local Area Network (WLAN) interworking security".

[3] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions given in TR 21.905 [3] and the following apply.

Application Specific Credentials: These are credentials e.g. keys, identifiers and related data, that are application specific and need to be protected against malicious access and only to be released to authorized applications acting as a Orpheus client. The application specific credentials might be stored or generated in the UICC or in the Persephone server. The application specific credentials can be generated from a master secret, randomly or be set by the user.

Charon Fine Grained Access Control: The access control policy in the terminal controls which authorized applications in the terminal have access to certain application specific application credentials, i.e., only certain application in the terminal is allowed to application specific credentials that can be used with certain applications.

Coarse-grained access control policy: In GAA context, the access control policy in the terminal controls whether an application is authorized to have access to NAF specific GAA credentials. Therefore, the application has access to all possible NAF specific GAA credentials. The coarse-grained access control policy may be stored in the UICC or in the ME.

Credential Generator: The credential generator generates the application specific credentials and the master secret. It might also generate directly the application specific credentials without a master secret. The Credential Generator might be part of an application or co-hosted together with an application.

Fine-grained access control policy: In GAA context, the access control policy in the terminal controls which authorized applications in the terminal have access to certain NAF specific GAA credentials, i.e., only certain application in the terminal is allowed to GAA credentials that can be used with certain NAFs. The fine-grained access control policy may be stored in the UICC or in the ME.

GAA Client: A software component in the terminal that communicates with a NAF in the network. The GAA client uses GAA credentials to authenticate and possibly otherwise secure the communication with the NAF in the network. GAA client uses the API provided by the GAA Server to gain access to the GAA credentials.

GAA Credentials: Consists of a bootstrapping transaction identifier (B-TID), one or two NAF specific keys, and a lifetime of those keys. If the terminal is equipped with GBA_U unaware UICC, then there is only one key (the GBA_ME NAF specific key) that derived by the GAA Server from the GAA master secret and given to the GAA client for further usage with the NAF. If the terminal is equipped with GBA_U aware UICC, then there are two keys that are derived from the GAA master secret in the UICC: one key (Ks_int_NAF) that stays and is used in the UICC, and one that is given to the ME and is used in the ME (Ks_ext_NAF similar to the GBA_U unaware key).

GAA Master Secret: GAA master secret Ks [1] is established during the bootstrapping session between the terminal (i.e., GAA Server for GBA_ME and UICC for GBA_U) and the BSF. The GAA master secret is used as a key to derive further NAF specific keys that can be used between the GAA clients and the NAFs. If the terminal is equipped with GBA_U unaware UICC, then the GAA Master secret is stored in the GAA server and GBA_ME is used. If the terminal is equipped with GBA_U aware UICC and GBA_U is used, then the GAA Master Secret is stored in the UICC.

GAA Master Secret stored in the GAA server corresponds to GAA Master Secret established with a terminal equipped with GBA_U unaware UICC (GBA_ME procedure used).

GAA Server: The software component in the terminal responsible for communicating with the SIM/USIM/ISIM application in the UICC, and with the BSF during bootstrapping procedure. The GAA server also functions as a public API towards the GAA clients in the terminal.

Master Secret: The Master Secret is a master secret that servers as a basis for later key derivations. The Master Secret is established between the terminal (i.e., Persephone Server) and the network. The master secret is used as a key to derive further application specific credentials that can be used between the Orpheus clients and the application. Not every application derives its credentials from a master secret.

Orpheus Client: A software component in the terminal that communicates with an application in the network. The Orpheus client application specific credentials to perform security functionalities. Orpheus Client uses the API provided by the Persephone Server to gain access to the Application Specific Credentials**.**

Persephone Server: The software component in the terminal responsible for communicating with the SIM/USIM/ISIM application in the UICC, and with external entities for possible key generation processes. The Persephone server also functions as a public API towards the Orpheus Clients in the terminal.

Styx Coarse Grained Access Control: The access control policy in the terminal controls whether an application is authorized to have access to application specific application credentials. Therefore, the application has access to all possible application specific credentials.

NOTE: The Greek Mythology was used in this Technical Report to minimize conflicts with other existing security specifications.

3.2 Abbreviations

The abbrevitations of TR 21.905 [3] apply, for the Generic Authentication Architecture (GAA) specific abbreviations we refer to [1] and for the I-WLAN specific abbreviations see [2]. In case of conflict, [1] and [2] take precedence.

4 Recommendations for trusted open platforms in 3GPP Release 7

4.1 Recommendations from the Generic Bootstrapping Architecture

4.1.1 Study of GBA in open trusted platforms

Figure 4-1 depicts the GAA related functionalities in the terminal. In relation to the GAA architecture, the GAA server communicates with the BSF server over Ub reference point, and with the UICC through the relevant device drivers, for example. The GAA client communicates over the network with a NAF server and the GAA server to obtain the NAF specific GAA credentials. When a NAF server requests a GAA client to authenticate itself with GAA credentials, the client communicates with the GAA server for GAA credentials specific to that NAF.

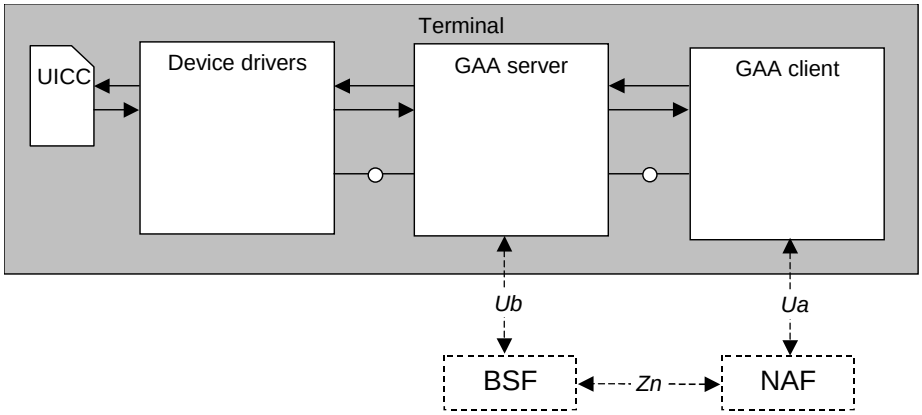

Figure 4-1 GAA related modules in terminal

The inherent feature of open platforms is that new applications can be installed to the terminal. In relation to GAA, this poses a security threat when a malicious application is installed in a type of terminal that does not protect GAA related functionalities. The GAA related security threats are as follows:

- A malicious application can access the UICC directly and therefore can function as a GAA server, which can then communicate with both the UICC and the BSF, and hence, establish the GAA master secret.
- The malicious application can access the GAA server private data, and gain access to the GAA master secret stored in the GBA_ME case, in the GAA server.
- The malicious application can access the GAA server API, and obtain NAF specific GAA credentials by requesting them from the GAA server.[1]

In all these cases, the malicious application can send either the GAA master secret that is stored in the GBA_ME case in the GAA server, or one or more NAF specific GAA credentials to the network. In this case, two attacks can be imagined: an attacker can masquerade as a NAF towards the terminal, or an attacker can masquerade as a UE towards a NAF. In both cases the attack can result a loss of private data, or unauthorized usage of the service. Also, the malicious application can itself masquerade as a UE towards a NAF, and gain access to the service provided by the NAF.

The first two threats can be mitigated by restricting access to the UICC, and the GAA server private data (see recommendations 1 and 2).

The third threat can be mitigated by restring the access to the GAA server API to authorized applications (see recommendation 3). The decision whether an application is authorized is done by the terminal manufacturer, the

[1] We don't explicitly consider the threat that a malicious application may steal NAF specific GAA credentials from another application, which has obtained them legitimately. We assume that the platform supports per-application protection of memory and storage.

operator, or the user (see recommendations 4 and 5). This can be called the **coarse-grained** access control method, where an authorized application has access to all possible NAF specific GAA credentials.

The coarse-grained access control method is enough for applications that have been authorized by the terminal manufacturer, or the operator.

However, if the user grants the authorization to the application, there may be a need for a **fine-grained** access control method. Users are not considered to be careful security wise, e.g., a user may grant the authorization to an application simply because an application just requests it. Therefore it may be required that the certain NAF specific GAA credentials are limited only to certain applications and user installed applications, even with GAA access authorization, would not have access to them. For example such credentials might give access to operator's NAF servers. Recommendation 6 addresses this issue.

To be useful, the fine-grained access control needs to be configured. First, the manufacturer or the operator may pre-configure the terminal (recommendation 7). Second, the operator may wish to update the configuration (recommendation 8). Finally, the user may add new policies to the configuration (recommendation 9). Note that a policy set by a user will not override a policy set by an operator or a manufacturer. However, the step two implies that an operator can override or modify a policy set by the manufacturer.

These are the recommendations identified to achieve the GAA security in the open platform terminals:

Table 4-1: Recommendations

ID	Recommandations	Commentas
1	It is possible for the platform to control access to the UICC.	Only authorized applications should have access to the UICC. Otherwise malicious application can perform the GAA server functionalities, establish the GAA master secret with the BSF server, send out the master secret (in case that the terminal is equipped with GBA_U unaware UICC), and thus generate all the GBA_ME NAF specific keys offline (outside the terminal), send out the received NAF specific key (Ks_ext_NAF) in case that the terminal is equipped with GBA_U aware UICC (online).
2	It is possible for the platform to restrict the access to GAA master secret of the GAA server.	If the GAA master secret in the GAA server in the case of GBA_ME is not protected, a malicious application can get access to it, send it out from the terminal, and the attacker can generate the GBA_ME NAF specific keys offline (outside the terminal).
3	It is possible for the platform to control general access in coarse-grained model to the GAA server so that an unauthorized application are not be able to get any NAF specific GAA credentials from the GAA server.	If a malicious application can gain access to the GAA server, it can generate all the NAF specific keys online (in the terminal) in case of GBA_ME and in the case of GBA_U it gains access to Ks_ext_NAF.
4	It is possible that an application is granted access to the GAA server by the manufacturer, or the operator.	The manufacturer or the operator must make sure that the application that is granted access to the GAA server is not malicious or have security flaws.
5	It is possible that an application is granted access to the GAA server by the user provided that such access is not prevented by manufacturer or operator policy. (See the next recommendation)	User may grant access to a malicious application simply because the application requests to have access.
6	In addition to recommendations 3, 4, and 5, it is possible to control the access to certain NAF specific GAA credentials in more fine-grained level, where access to certain NAF specific GAA credentials can be restricted to certain applications only.	This recommendation can protect against malicious applications that try to get access to certain NAF specific GAA credentials.
7	In addition to recommendation 6, the manufacturer can pre-configure the fine-grained access control policy on the terminal or the operator can pre-	

	configure the fine-grained access control policy on the terminal or UICC.	
8	In addition to recommendation 6, the operator can update all fine-grained access control policies on the terminal or the UICC.	
9	In addition to recommendation 6, the user can add new fine-grained access control policies to the terminal or the UICC.	The user may only add and modify user's own policies. The user cannot change policies set by the manufacturer or the operator.

The recommendations in Table 4-1 can be divided in to three groups the following way:

- **Group 1**: To provide the basic GAA related security in the terminal, recommendations 1 to 4 must be enforced. The basic GAA security includes controlling access to UICC and to GAA master secret, controlling access to the GAA server, and only the manufacturer or the operator can grant access to the GAA functionality for the application.
- **Group 2:** If access to GAA functionality for applications can be granted by the user then the recommendation 5 must be enforced.
- **Group 3:** If more fine-grained access to the GAA functionality is wanted as described above, then recommendations 6 to 8 must be enforced. Recommendation 9 must be enforced if the user can grant access to GAA functionality for an application

If there are applicable access control policies stored in the UICC and the terminal, then access control policies stored on the UICC should take precedence over access control policies stored in the terminal. If the user is not allowed to grant access to GAA functionality for an application, it is enough to enforce the recommendation group 1. In this case, only the manufacturer or the operator can have granted access to GAA functionality for an application, and therefore it should be assumed that the application is trusted.

If the user is allowed to grant access to GAA functionality for an application, then naturally both recommendation groups 1 and 2 need to be enforced. In this case, the user is allowed to grant access to GAA functionality and user may do this for any application that requests this access. It should be assumed that a malicious application might thus gain access to GAA functionality in the terminal.

If the user is allowed to grant access to GAA functionality for an application, then for added protection, the recommendation group 3 should be enforced as well. This would enable GAA functionality to protect certain NAF specific GAA credentials that have more value -- such as NAFs that belong to the operator.

Another alternative is that the GAA server will show a dialog window to the user whenever an application (that does not have access to GAA functionality granted by the manufacturer or the operator) attempts to gain access to GAA functionality. The dialog would show the application name, the NAF name (i.e., the FQDN of the NAF server), and ask whether this application is allowed to gain access to the GAA credentials for this particular NAF. The user would have the choice of either granting or denying access. The dialog could also offer to remember the decision made by the user, and the GAA server would remember this decision for the next time the same application requests the same NAF specific GAA credentials.

User interaction on security is a common source of misunderstandings, hence user interaction should be minimized to obtain a good usage experience and minimize security risks. The access to the GAA functionality and sensitive GAA material should only be granted by operators or manufacturers. Third parties could have contractual relationships with operators or manufactures and therefore could obtain access to the GAA functionality using this relationship using digital signatures.

The trust an NAF or the terminal application can put into the NAF-specific credentials depends on the underlying security baseline used for the generation and storage of Ks_(ext/int)_NAF. In an open trusted platform, and also in a closed platform, the application in the terminal might not be aware what kind of security baseline has been used and therefore is not able to enforce an security level information in the client application in the terminal. In principle, the NAF can receive this information from the BSF and stop service delivery when the security level is not sufficient. But to be able to do this, first the UE contacts the NAF, the NAF contact the BSF, the BSF contacts the HSS and then returns the information to the NAF. This may result in the fact that the user pays for the contact to the NAF, without being able to obtain the service and generates network load, without any kind of revenue generation. The UE may have information on the required security level e.g. received in form of a flag or preconfigured.

The GAA server in the UE has knowledge about the security baseline used for the generation and storage of Ks_(int/ext)_NAF, but if the GAA client application has information on the security requirements from the network, it can not enforce it without obtaining information from the GAA server on the actually used security method. The application in the terminal requests the GBA specific credentials from the GBA server. The response contains the credentials and the underlying security method used e.g.

- USIM with GBA_U aware UICC
- USIM without GBA_U aware UICC
- ISIM with GBA_U aware UICC
- ISIM without GBA_U aware UICC
- Legacy GBA

The application in the terminal can then process this information further, depending on the used Ua protocol, e.g. comparing the received information that may be received in form of a flag over the Ua reference point with configured security level information.

4.1.2 Recommendation

This technical report recommends that only recommendation group 1 in the previous clause is required. As user interaction on security is not desired and may cause breaches in GAA terminal security group 2 should be not be required. Recommendation group 3 is not needed as group 2 was excluded.

4.2 Recommendations from I-WLAN

The threats and recommendations described in 4.1 are also valid for I-WLAN. In I-WLAN there is a UE functionality split case, this case is also called split terminal. In this case the UICC/SIM card may reside in a 3GPP ME (acting as a (U)SIM "server") and be accessed by a WLAN-UE (acting as the (U)SIM "client") through local link (e.g. Bluetooth, Infrared or USB), and has the following additional requirements from [2]:

- Whenever someone tries to remotely access a (U)SIM some sort of alert shall be sent, e.g. a message shall be displayed informing the user of the attempted access and guiding him to choose "Allow", or "Disallow". The user can then decide whether the access is authorized or not and can opt for allow or disallow the access.
- The UICC holding device shall be responsible for scheduling all (possibly concurrent) accesses to the UICC by itself, and by one additional device connected via the local interface.
- Applications/Data information could be retrieved from (U)SIM, provided that the UICC (or SIM card) is inserted in a 3GPP ME. When the (U)SIM is re-used over local interfaces, further access control on the Applications/Data information shall be applied by the 3GPP ME holding the (U)SIM.

The new applications can be installed to the terminals (3GPP ME or WLAN-UE). This poses a security threat when a malicious application is installed in a type of terminal that does not protect I-WLAN related functionalities correctly. The related security threats are as follows:

- A malicious application can trigger the UICC and get the (Kc, IK, CK, MSK) from the UICC.
- The malicious application can access the copy of the keys (Kc, IK, CK, MSK) stored in ME.

In all these cases, the malicious application can send either the (Kc, IK, CK, MSK) secret, or one or more specific credentials to the network. In this case, two attacks can be imagined: an attacker can masquerade as a network element towards the terminal, or an attacker can masquerade as a UE towards the network. In both cases the attack can result a loss of private data, or unauthorized usage of the service.

These threats can be mitigated by restricting access to the UICC, and the 3GPP ME private data (see recommendations 1 and 2).

These are the recommendations identified to achieve the I-WLAN security in the open platform terminals:

Table 4-2: Recommendations

ID	Recommendations	Comments
1	It is possible for the platform to control access to the UICC.	Only authorized applications should have access to the UICC.
2	It is possible for the platform to restrict the access to the keys stored in the 3GPP ME.	If the keys in the 3GPP ME is not protected, a malicious application can get access to it and send it out from the terminal.
3	It is possible that an application is granted access to the keys in 3GPP ME by the manufacturer, or the operator.	The manufacturer or the operator must make sure that the application that is granted access to the keys in the 3GPP ME is not malicious or have security flaws.
4	It is possible that an application is granted access to the keys in 3GPP ME by the user provided that such access is not prevented by manufacturer or operator policy.	User may grant access to a malicious application simply because the application requests to have access. The access to the keys stored in 3GPP ME can be restricted to certain applications only.
5	The manufacturer can pre-configure the access control policy on the terminal or the operator can pre-configure the access control policy on the terminal or UICC.	
6	The operator can update all access control policies on the terminal or the UICC.	
7	The user can add new access control policies to the terminal or the UICC.	The user may only add and modify user's own policies. The user cannot change policies set by the manufacturer or the operator.

The recommendations in Table 4-2 can be divided in to three groups the following way:

- **Group 1**: To provide the basic security in the terminal, recommendations 1 to 4 must be enforced. The basic security includes controlling access to UICC and the (Kc, IK, CK, MSK) used in I-WLAN, controlling access to the derived secrets, and only the manufacturer or the operator can grant access to the secrets for the application.
- **Group 2:** If access to keys for applications can be granted by the user then the recommendation 4 must be enforced.
- **Group 3:** If more fine grained access control is wanted as described above, then recommendations 5 and 6 must be enforced. Recommendation 7 must be enforced if the user can grant access to keys for an application.

If the user is not allowed to grant access to keys for an application, it is enough to enforce the recommendation group 1. In this case, only the manufacturer or the operator can have granted access to keys for an application, and therefore it should be assumed that the application is trusted.

If the user is allowed to grant access to keys for an application, then naturally both recommendation groups 1 and 2 need to be enforced. In this case, the user is allowed to grant access to keys and user may do this for any application that requests this access. It should be assumed that a malicious application might thus gain access to keys in the terminal.

If the user is allowed to grant access to keys for an application, then for added protection, the recommendation group 3 may be enforced as well.

User interaction on security is a common source of misunderstandings, hence user interaction should be minimized to obtain a good usage experience and minimize security risks. The access to the UICC/ SIM and keys stored in the 3GPP ME should only be granted by operators or manufacturers. The user interaction on security is required is specified in TS 33.234 [2].

4.3 Generalized recommendations

4.3.1 Study of credential security in open trusted platforms

The inherent feature of open platforms is that new applications can be installed to the terminal. In relation to shared secret credential management, this poses a security threat when a malicious application is installed in a type of terminal that does not protect the shared secret properly. The security threats are as follows:

- A malicious application can access the UICC directly and therefore can then communicate with both the UICC and a credential generator and hence establish master secrets and application specific credentials.
- The malicious application can access the Persephone server private data, and gain access to the master secret.
- The malicious application can access the Persephone server API, and obtain the application specific credentials by requesting them from the Persephone server.

In all these cases, the malicious application can send either the master secret, or one or more application specific credentials to the network.

NOTE1: If there are only application specific credentials e.g. password storage or just one application using a master secret, then the second threat can be neglected.

The first two threats can be mitigated by restricting access to the UICC, and the Persephone Server private data (see recommendations 1 and 2).

The third threat can be mitigated by restring the access to the Persephone Server API to authorized applications (see recommendation 3). The decision whether an application is authorized is done by the terminal manufacturer, the operator, or the user (see recommendations 4 and 5). This can be called the **Styx Coarse Grained Access Control** access control method, where an authorized application has access to all possible application specific credentials.

The **Styx Coarse Grained Access Control** access control method is a basic access control that is enough for applications that have been authorized by the terminal manufacturer, or the operator.

However, if the user grants the authorization to the application, there may be a need for a more granulated access control method. i.e. **Charon Fine Grained Access Control.** It may be required that the certain application specific credentials are limited only to certain applications and user installed applications, even with credential access authorization, would not have access to them. For example such credentials might give access to operator's application servers. Recommendation 6 addresses this issue.

To be useful, the Charon fine grained access control needs to be configured. First, the manufacturer or the operator may pre-configure the terminal (recommendation 7). Second, the operator may wish to update the configuration (recommendation 8). Finally, the user may add new policies to the configuration (recommendation 9). Note that a policy set by a user will not override a policy set by an operator or a manufacturer. However, the step two implies that an operator can override or modify a policy set by the manufacturer.

These are the recommendations identified to achieve the credential security in the open platform terminals:

Table 4-5: Recommendations

ID	Recommandations	Commentas
1	It is possible for the platform to control access to the UICC.	Only authorized applications should have access to the UICC. Otherwise malicious application can establish a master secret, depending on storage and access mechanisms having access to application specific credentials or, in case of master secret stored outside the UICC, being able to generate application specific credential offline (outside the terminal) .
2	It is possible for the platform to restrict the access to master secret of the Persephone server.	If the master secret of the Persephone server is not protected, a malicious application can get access to it, send it out from the terminal, and the attacker can generate all the application specific keys offline (outside the terminal). The attack does not apply in

		case of master secret stored in the UICC.
3	It is possible for the platform to control general access in coarse-grained model to the Persephone server so that an unauthorized application is not be able to get any application specific credentials from the Persephone server.	If a malicious application can gain access to the Persephone server and the master secret is stored there, then it can generate all the application specific keys online (in the terminal). In case of master secret stored on the UICC the attacker cannot generate application keys derived from the master secret.
4	It is possible that an application is granted access to the Persephone server by the manufacturer, or the operator.	The manufacturer or the operator must make sure that the application that is granted access to the Persephone server is not malicious or have security flaws.
5	It is possible that an application is granted access to the Persephone server by the user provided that such access is not prevented by manufacturer or operator policy. (See the next recommendation)	User may grant access to a malicious application simply because the application requests to have access.
6	In addition to recommendations 3, 4, and 5, it is possible to control the access to certain application specific credentials in more fine-grained level, where access to certain application specific credentials can be restricted to certain applications only.	This recommendation can protect against malicious applications that try to get access to certain application specific credentials.
7	In addition to recommendation 6, the manufacturer can pre-configure the fine-grained access control policy on the terminal or the operator can pre-configure the fine-grained access control policies on the UICC.	
8	In addition to recommendation 6, the operator can update all fine-grained access control policies on the terminal or UICC.	
9	In addition to recommendation 6, the user can add new fine-grained access control policies to the terminal or UICC.	The user may only add and modify user's own policies. The user cannot change policies set by the manufacturer or the operator.

The recommendations in Table 4-5 can be divided in to three groups the following way:

- **Group 1**: To provide the basic application credential related security in the terminal, recommendations 1 to 4 must be enforced. The basic credential security includes controlling access to UICC and to the master secret, controlling access to the Persephone server, and only the manufacturer or the operator can grant access to the credential functionality for the application.

- **Group 2:** If access to credential functionality for applications can be granted by the user then the recommendation 5 must be enforced.

- **Group 3:** If more fine-grained access to the credential functionality is wanted as described above, then recommendations 6 to 8 must be enforced. Recommendation 9 must be enforced if the user can grant access to credential functionality for an application

NOTE2: If there is only an application specific secret and no master secret, then the recommendations are not impacted. If there are only a master secret, that is used by several applications, then there can be no fine-grained access control. If some applications utilize a master secret for key derivation and other applications have their own non-master secret based credentials, then the same recommendations apply as outlined above.

If there are applicable access control policies stored in the UICC and the terminal, then access control policies stored on the UICC should take precedence over access control policies stored in the terminal. If the user is not allowed to grant access to credential functionality for an application, it is enough to enforce the recommendation group 1. In this case, only the manufacturer or the operator can have granted access to credential functionality for an application, and therefore it should be assumed that the application is trusted.

If the user is allowed to grant access to credential functionality for an application, then naturally both recommendation groups 1 and 2 need to be enforced. In this case, the user is allowed to grant access to credential functionality and user may do this for any application that requests this access. It should be assumed that a malicious application might thus gain access to credential functionality in the terminal.

If the user is allowed to grant access to credential functionality for an application, then for added protection, the recommendation group 3 should be enforced as well. This would enable credential functionality to protect certain application specific credentials that have more value, such as application servers that belong to the operator.

4.3.2 Recommendation

This technical report recommends that only recommendation group 1 in the previous clause is required. As user interaction on security is not desired and may cause breaches in credential terminal security group 2 should be not be required. Recommendation group 3 is not needed as group 2 was excluded.

Annex A:
Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2005-10					Creation of document		0.0.0
2005-11					Integration of pseudo-CR S3-050717 of SA3#41 Meeting	0.0.0	0.1.0
2006-02					Integration of pseudo-CR S3-060075, S3-060085 and S3-060086 of SA3#42 meeting (output number S3-060177)	0.1.0	0.2.0
2006-04					Integration of discussion results of SA3#42 meeting S3-060255	0.2.0	0.3.0
2006-04					Integration of S3-060256 at SA3#43	0.3.0	0.4.0
2006-06	SP-32	SP-060433			For information to SA#32	0.4.0	1.0.0
2006-08					Integration of S3-060548	1.0.0	1.1.0
2006-11					Integration of pseudo-CRs S3-060642 and S3-060643 from SA3#45 (output number for TR S3-060827)	1.1.0	1.2.0
2007-02					Integration of Pseudo-CR S3-070310	1.2.0	2.0.0
2007-03	SP-35	SP-070165			Approved	2.0.0	7.0.0
2008-12	SP-42	--	--	--	Upgrade to Release 8	7.0.0	8.0.0
2009-12	-	-	-	-	Update to Rel-9 version (MCC)	8.0.0	9.0.0
2011-03	-	-	-	-	Update to Rel-10 version (MCC)	9.0.0	10.0.0
2012-09	-	-	-	-	Update to Rel-11 version (MCC)	10.0.0	**11.0.0**
2014-09	-	-	-	-	Update to Rel-12 version (MCC)	11.0.0	**12.0.0**

History

Document history		
V12.0.0	October 2014	Publication