Open Trusted Computing

Royal Holloway
University of London

Trusted Mobile Platforms:

Part 2: Trusted computing in mobile devices

Chris Mitchell
Royal Holloway, University of London
c.mitchell@rhul.ac.uk
http://www.isg.rhul.ac.uk/~cjm

Contents

- TCG and the MPWG
- Implementing OMA DRM v2
- SIMLock using trusted computing
- Secure software download
- Concluding remarks

Contents

- TCG and the MPWG
- Implementing OMA DRM v2
- SIMLock using trusted computing
- Secure software download
- Concluding remarks

Development of Trusted Mobile Platforms

- Trusted computing technology is becoming common in new PCs, at least as far as TPMs is concerned.
- Situation not so advanced for other types of platform.
- While there are many potential applications for TC in mobile devices (e.g. PDAs, smart phones, ...), TPMs are not yet included in such platforms.
- For many reasons (cost, complexity, ...) TC may be implemented in rather different ways in mobile devices.
- Mobile devices may not include a TPM, but instead TPM functionality could be implemented using a combination of trusted hardware functionality in a mobile platform and software.
- How this works will probably vary widely.

The MPWG of the TCG

- The TCG provides specifications for any type of device.
- Initial standardisation work focused on the TPM and a standard set of TPM APIs for software developers/vendors.
- More recently, baseline TCG specification set has been expanded by platform-specific working groups (WGs) to give specifications for specific platform implementations (e.g. for PC clients, servers, peripherals and storage systems).
- The Mobile Phone Working Group (MPWG) is one such WG.
- It has developed a set of specifications for what TC functionality should be supported by a mobile platform.
- This has been developed as a result of the analysis of a series of mobile use cases.

MPWG activity

- The main challenge for the MPWG is to determine the ‘roots of trust’, needed in a trusted mobile phone.
- To identify the capabilities needed by a trusted mobile phone, a number of use cases have been identified.
- These include: SIMLock, device authentication, mobile ticketing, mobile payment and robust DRM implementation.

MPWG use cases

- The use cases enable the MPWG to:
  - derive requirements that address the situations described in the use cases;
  - specify an architecture based on the TCG architecture that meets these requirements; and
  - specify the functions and interfaces that meet the requirements in the specified architecture.

MTM specifications

- MPWG has recently published the **TCG Mobile Trusted Module (MTM) Specification**.
- A mobile platform will typically contain multiple MTMs to support multiple mobile device stakeholders.
- It is envisaged that each MTM will provide a subset of the TPM v1.2 functionality.
- Some MTMs may also contain additional functionality to ensure that parts of the device boot into a preset state (i.e. secure boot functionality).
- Two types of MTM have been defined.

The MLTM

- A **Mobile Local-owner Trusted Module** (MLTM) supports uses (or a subset of uses) similar to those of existing v1.2 TPMs, where the device is controlled by an entity with physical access to the platform.
- Some TPM v1.2 functionality may not be supported because of the restrictions in today’s phone technologies.
- The use cases described by the TCG have been analysed to determine the subset of functionality required within a MTM to enable their secure implementation.

The MRTM

- A **Mobile Remote-owner Trusted Module** (MRTM) also supports a subset of uses similar to those of existing v1.2 TPMs.
- It moreover enables a remote entity (such as the device manufacturer or network operator) to predetermine the state into which some parts of the phone must boot.

Mobile applications

- The applications for trusted mobile phones discussed in the current TCG MPWG use case document cover:
  - the protection of downloaded content and software;
  - the protection of user data and identity information, and service identity information; and
  - enabling mobile payment and mobile ticketing.
- In this talk we examine three typical applications from the first two of the above three categories. [Detailed analyses of these applications have been carried out by RHUL within the OpenTC project].

Contents

- TCG and the MPWG
- Implementing OMA DRM v2
- SIMLock using trusted computing
- Secure software download
- Concluding remarks

OMA

- Open Mobile Alliance (OMA) was founded in June 2002.
- One of the original objectives of the OMA was to define a DRM specification set for use in the mobile environment.
- OMA DRM v1 was published as a candidate specification in October 2002, and in 2004 was approved as an OMA enabler specification after full interoperability testing had been completed.
- Following this, in 2004, work on OMA DRM v2 was completed and OMA DRM v2 was published as a candidate specification in July 2004.
- OMA DRM v2 builds upon the version 1 specifications to provide higher security and a more extensive feature set.

OMA DRM v1

- Main goals:
  - Timely and inexpensive to deploy;
  - Easy to implement on mass market mobile devices;
  - It was required that the initial OMA DRM solution did not necessitate the roll-out of a costly infrastructure.
- Three defined classes of DRM functionality:
  - Forward lock (where forwarding of content is prohibited);
  - Combined delivery (of content and rights);
  - Separate delivery (of content and means to access the content).

OMA DRM v1

- Weaknesses in OMA DRM v1:
  - A rights issuer has no way in which to determine whether the requesting device supports DRM;
  - In the separate delivery DRM class, where the content is encrypted, the content encrypting key is not protected;
  - The device has no way of authenticating the rights issuer and therefore may be sent bogus rights objects from an entity claiming to be the legitimate rights issuer.

OMA DRM v2

- Both device authentication and rights issuer (RI) authentication are provided.
- Mechanisms are deployed in order to protect the confidentiality of media objects.
- Mechanisms are also deployed so that the OMA DRM v2 agent can determine whether a media object received from an RI has been modified in an unauthorised way.
- Also supports an extended feature set: subscription, streaming content, reward schemes, domains, unconnected devices.
- Note that we use the term ‘agent’ to describe the software within the mobile device that enforces the OMA DRM rules (this terminology is also used in our other use case descriptions).

Security-critical data

- OMA DRM agents must be equipped with the necessary security-critical data.
- Every OMA DRM v2 agent is assigned a unique key pair.
- The private key from this key pair is used by an OMA DRM v2 agent to generate digital signatures, so a rights issuer can authenticate a particular DRM agent.
- The public key is used by rights issuers to distribute rights object encryption keys, which are used to protect content encryption keys, that are themselves used to encrypt content.
- [Note that this means that the same key pair is used for signature and encryption – not good practice].

Certificates

- A **DRM agent certificate** is provided to the DRM agent; this certificate binds the agent to its public key.
- The certificate can be specified as part of one or more certificate chains.
- In such a chain, the OMA DRM v2 agent certificate comes first, and each subsequent certificate contains the public key necessary to verify the certificate preceding it.
- The RI indicates its preferred trust anchor(s), i.e. its trusted root CA(s), and the OMA DRM v2 agent must send back a device certificate (chain) which points to an appropriate anchor. This enables the RI to verify the OMA DRM v2 agent certificate.

Other security data

- The **device details** indicate the device manufacturer, model, and version number.
- The **trusted RI authorities certificate** is used to indicate which rights issuer trust anchor(s) are recognised by the OMA DRM v2 agent.
- This trusted RI authorities certificate may either be:
  - a single root certificate, as is the case in the CMLA trust model where the trusted RI authorities certificate is a self-signed CMLA root CA certificate, or
  - a collection of self-signed public key certificates.

Supported security features

- The **Rights Object Acquisition Protocol** (ROAP) suite:
  - The 4-pass registration protocol;
  - The 2-pass rights acquisition protocol;
  - The 1-pass rights acquisition protocol;
  - The 2-pass join domain protocol;
  - The 2-pass leave domain protocol.
- A trust model enables an RI to get assurance about DRM agent behaviour and the robustness of the DRM agent implementation:
  - It is the responsibility of the **Content Management Licensing Administrator** (CMLA), or a similar organisation, to provide a trust model, i.e. robustness rules, and to define actions which may be taken against a manufacturer who builds devices which are not sufficiently robust.

Mapping to TCG functionality

- All the OMA DRM v2 functions have been mapped to TCG functionality.
- That is, a detailed set of TPM commands have been identified which will robustly support all the OMA DRM v2 features.
- This analysis has contributed to the development of a mobile profile of TCG functionality.
- We now briefly review some of this analysis.

Using secure boot I

- TC cannot guarantee the integrity of the OMA DRM v2 agent while stored; however, TC mechanisms can be used to help detect malicious or accidental modifications/removal.
- Secure boot functionality can be used to ensure that security-critical platform components boot into a predetermined state.
- As discussed in Part I, secure boot is not supported by the TCG TPM main specifications.
- However, work on secure boot has been conducted independently of the TCG by many authors.
- They all describe a similar process: system components are measured, and the results are compared with a set of (securely stored) expected values, accessed by the platform during boot.

Using secure boot II

- If, at any stage during the boot process, the removal or modification of a platform component, e.g. the OMA DRM v2 agent, is detected, the boot process is aborted.
- While secure boot is not specified in the TPM specification set, the TCG mobile phone working group Mobile TPM specification does include a secure boot process.
- Security-critical data associated with the OMA DRM v2 agent, such as the device details and the trusted rights issuer authorities certificate (specifying which CAs are trusted by the rights issuer), which require integrity protection while in storage, can also be verified as part of a secure boot.

Using sealed storage I

- Alternatively, sealed storage functionality can be used in order to detect the malicious or accidental modification or removal of the OMA DRM v2 agent while in storage, and, indeed, to store data which needs to be confidentiality and/or integrity-protected.
- It can also ensure that sensitive data is only accessible by authorised entities when the mobile device is in a predefined state, for example, when a legitimate OMA DRM v2 agent is executing in an isolated execution environment.
- The security-critical data and any domain and RI context information to be protected is associated with a ‘digest at creation’ and a ‘digest at release’, and encrypted by the TPM.
- The ‘digest at release’ means that the platform must be in the corresponding state to access the data.

Using sealed storage II

- While integrity protection is not explicitly provided, 20 bytes of authorisation data are associated with the data to be sealed prior to encryption, giving integrity-protection for the stored data.
- The sealed data is asymmetrically encrypted and the corresponding private decryption key is securely stored in the TPM, giving confidentiality-protection for the stored data.
- Including 20 bytes of authorisation data and the digest at release with the sealed data before encryption ensures that only an authorised entity can access the data, and access can only take place when the platform is in the required software state.
- Finally, sealing the data to a specified platform configuration also ensures that any unauthorised modification and/or removal of security-critical software (e.g. the OMA DRM v2 agent) reflected in the digest at release will be detected, and access to the sealed data denied.

Key management

- Not only can the TPM confidentiality and integrity-protect the OMA DRM v2 private key, the TPM can also be used to generate the required OMA DRM v2 agent asymmetric key pair as well as to protect the private key while stored and in use.
- TC functionality also enables the isolation of security-critical software and data in a secure execution environment so that it cannot be observed and/or modified in an unauthorised manner by software executing in parallel execution environments.
- A good quality random number generator is provided by a TPM, enabling the generation of unpredictable nonces for use in the ROAP suite protocols, preventing replay and preplay attacks.
- The TPM can also support accurate time synchronisation.

Contents

- TCG and the MPWG
- Implementing OMA DRM v2
- SIMLock using trusted computing
- Secure software download
- Concluding remarks

What is SIMLock?

- Mobile device personalisation, or SIMLocking, enables a device to be forced to operate only with certain (U)SIMs.
- SIM mobility has many advantages – however, there are also disadvantages.
- Phone operators who subsidise mobile equipment, and hope to recover this initial loss from future profits from subscriptions, lose if mobile device users can move a phone to another network before the original subscription has finished.
- SIM mobility may also encourage handset theft for re-use or re-sale.
- These issues have led to the development of SIMLock.

Personalisation categories

- SIMLock has five personalisation categories, i.e. five ways of controlling use of device:
  - **Network** – network operator personalises a device so it can only be used with (U)SIMs from that operator;
  - **Network subset** – network operator personalises a device so it can only be used with certain (U)SIMs from that operator;
  - **Service provider** – service provider personalises a device so it can only be used with (U)SIMs from that service provider;
  - **Corporate** – company personalises an employee/customer device so it can only be used with (U)SIMs owned by that company; and
  - **SIM/USIM** – end user personalises a mobile device so it can only be used with a particular (U)SIM.

Security-critical data

- A personalisation indicator and a personalisation code or code group are associated with each personalisation category:
  - a **personalisation indicator** is used to specify whether a personalisation category is active (‘on’) or deactivated (‘off’); each category has an independent indicator; if an indicator is active it means that the SIM has been locked to a network(s), network subset(s), service provider(s), corporate entity/entities or SIM/(U)SIM(s).
  - a **personalisation code** or **code group** is used to indicate how that personalisation indicator controls device operation, i.e. it specifies the network, the network subset, etc.; an independent personalisation code or code group is defined for each category.

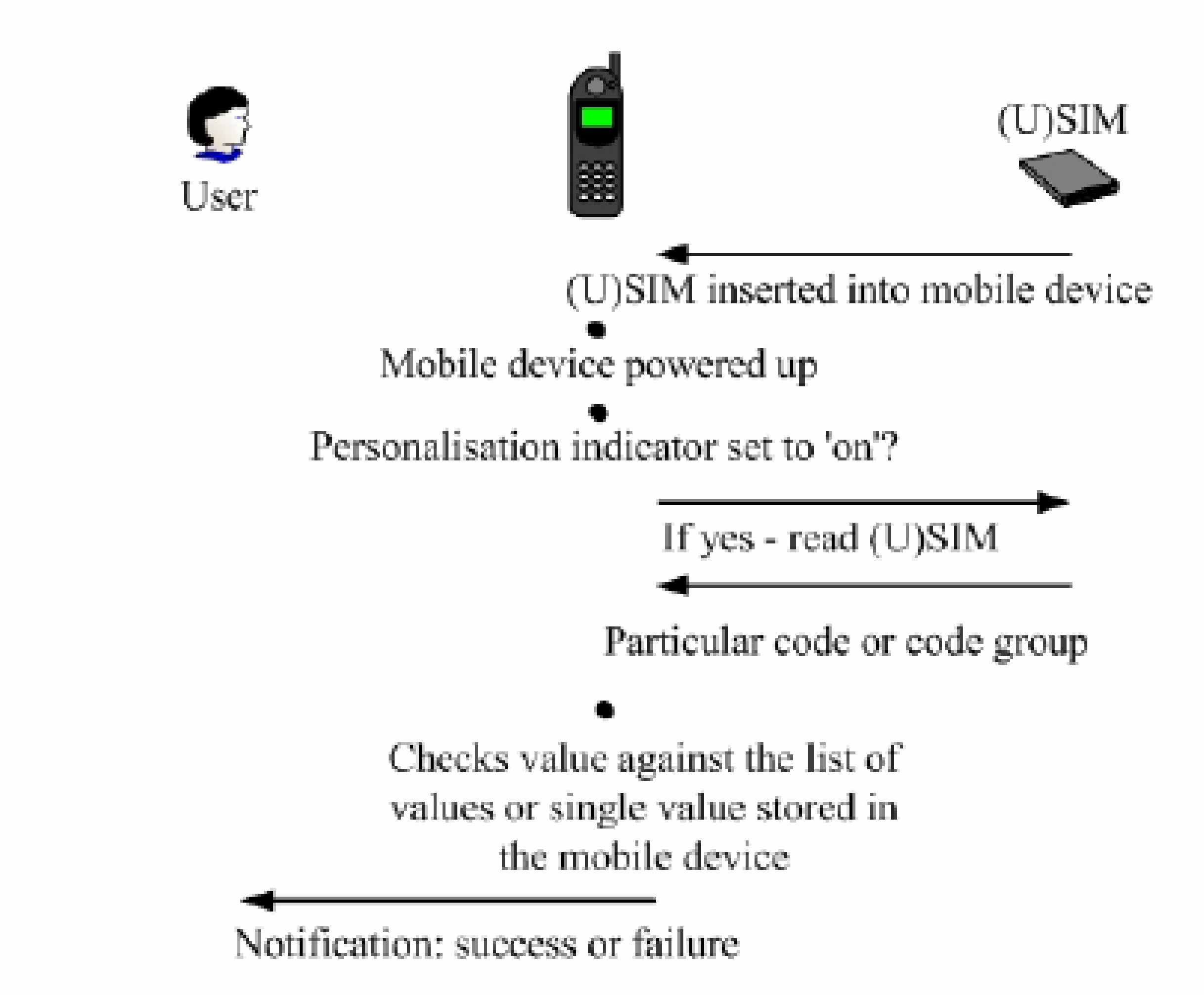
Controlling SIMLock

- To personalise a device, the required personalisation code or code group must be entered into the device and the appropriate personalisation indicator set to ‘on’.
- The relevant control key, used for device de-personalisation, must be also be stored within the device (entry of the key is necessary to switch personalisation off).

SIMLock operation

- When a (U)SIM is inserted into the device, or when the device is powered on, the mobile device checks which personalisation indicators are set to ‘on’.
- The personalisation agent reads the (U)SIM, and extracts the required code(s)/code group(s).
- The code(s)/code group(s) are then verified against the list of values stored on the mobile device.
- The mobile device then responds accordingly, displaying a message of success or failure to the device user.
- Should this checking process fail, the device enters a ‘limited service state’ in which only emergency calls can be made.

SIMLock operation



User

(U)SIM

(U)SIM inserted into mobile device

Mobile device powered up

Personalisation indicator set to 'on'?

If yes - read (U)SIM

Particular code or code group

Checks value against the list of values or single value stored in the mobile device

Notification: success or failure

De-personalisation

- To de-personalise a device, the control key for the particular personalisation category must be entered into the device.
- This is compared against the control key stored in the device.
- If the entered control key matches the stored value, then the personalisation indicator for the category in question is set to ‘off’.

Threat analysis

- Threats to the SIMLock process include the following:
  - Unauthorised modification or removal of the device personalisation agent software while in storage on or while executing on the device.
  - Unauthorised reading/copying of a control key while in storage or in use on the device.
  - Unauthorised modification or deletion of a personalisation code/code group, control key or personalisation indicator while in storage or in use on the device.

Using TC to support SIMLock I

- Unauthorised modification or removal of a device personalisation agent cannot be prevented using TC.
- However, while software is in storage, secure boot functionality can be used so that, at start-up, a measurement of the agent software is verified against an expected value.
- This enables unauthorised modification and/or removal to be detected.
- Security-critical data requiring integrity protection, e.g. network, network subset, corporate, and service provider codes/code groups and indicators, can also be covered by the secure boot.
- TC isolation mechanisms can be used to ensure the integrity of the personalisation agent, and that any security-critical data is protected while in use on the device.

Using TC to support SIMLock II

- Alternatively, personalisation code/code groups, indicators and control keys could simply be sealed to an isolated execution environment which hosts a device personalisation agent.
- In this way, security-critical data can be both integrity and confidentiality-protected while in storage.
- If the agent and/or the supporting environment to which the data is sealed are modified, then security-critical data will be inaccessible.
- While sealing ensures that data is released into a predefined execution environment, isolation technologies are needed to ensure that both the agent and the security related data are confidentiality and integrity-protected while in use on the platform.

Contents

- TCG and the MPWG
- Implementing OMA DRM v2
- SIMLock using trusted computing
- Secure software download
- Concluding remarks

Applications

- Two types of software can be downloaded to a mobile device, namely *application software* (e.g. games) and *core software* (e.g. operating systems software).
- We consider here secure download of core software.
- We identify two distinct cases of core software download:
  - **Software Defined Radio** (SDR) – SDR enables reconfiguration of a radio via software; it is clearly vital that such software is delivered in ways which guarantee integrity and origin (for safety reasons);
  - **Digital Video Broadcast** (DVB) to a mobile, could benefit from software download of conditional access systems; such systems are typically both confidentiality and integrity sensitive.

Application 1: SDR

- A software defined radio is a communications device “whose operational modes and parameters can be changed or augmented, post manufacturing via software” [SDR Forum].
- Devices can be reconfigured to communicate using multiple frequency bands and protocols, or upgraded at low cost.
- SDR is an important innovation for the communications industry, and provides many advantages over purely hardware-based terminals.
- Cost reductions may result from deployment of a generic hardware platform which can be customised.
- However, there are also major security threats and safety issues.

SDR Threat Analysis I

- Threats to the security of the downloaded radio software:
  - Unauthorised reading of radio software while in transit between the software provider and the end host, or while in storage or executing on the end host.
- *Impacts:*
  - Unauthorised access and execution of radio software;
  - Infringement of intellectual property rights;
  - Undesired reverse engineering of software.

SDR Threat Analysis II

- Threats to host security:
  - malicious or accidental modification or removal of security-critical software while in storage or executing on the end host.
  - malicious or accidental modification, addition or removal of downloaded radio software while in transit between the software provider and the end host, or while in storage or executing on the end host.
  - the download and execution of inappropriate radio software which does not meet the capability requirements of the SDR device.
- Impacts:
  - an inoperable device (e.g. improper change of modulation format);
  - violation of RF spectrum resulting in RF interference or user safety issues with spurious emissions;
  - increased output power resulting in safety issues and decreased battery life;
  - compromise of user data or applications.

Addressing threats to SDR using TC (protecting the radio software)

	Attestation	Protected storage	Isolation	Secure boot	Onus on software provider
Unauthorised reading of radio software:					
In transit		X (preventative)			
In storage		X (preventative)			
While executing			X (preventative)		

Addressing threats to SDR using TC (protecting the host)

	Attestation	Protected storage	Isolation	Secure boot	Onus on software provider
Unauthorised addition or modification of radio software:					
In transit					X (digital signature - preventative)
In storage					X (digital signature - preventative)
While executing			X (preventative)		

Addressing threats to SDR using TC (protecting the host)

	Attestation	Protected storage	Isolation	Secure boot	Onus on software provider
Unauthorised modification or removal of security critical software:					
In storage				X (detection)	
While executing			X (preventative)		
Inappropriate download	X (preventative)				

TC – Lessening threat impact

Impact – malicious software running Caused by:	Attestation	Protected storage	Isolation	Secure boot
Unauthorised addition or modification of radio software	X (reactive – reconnection to commercial network may be denied)	X (reactive – lessen impact on user data which has been sealed to trusted platform state)	X (reactive – lessen impact on other software running on the platform if it is isolated from malicious software)	X (reactive - ensure security-critical software is running correctly – e.g. filters)

TC – Lessening threat impact

Impact – malicious software running Caused by:	Attestation	Protected storage	Isolation	Secure boot
Unauthorised modification or removal of security critical software	X (reactive – reconnection to commercial network may be denied)	X (reactive – lessen impact on user data which has been sealed to trusted platform state)	X (reactive – lessen impact on other software running on the platform if it is isolated from malicious software)	
Inappropriate download	X (reactive – reconnection to commercial network)			X (reactive - ensure security-critical software is running correctly – e.g. filters)

Summary

- Trusted computing mechanisms:
  - Authenticated boot;
  - Protected storage;
  - Attestation;
  - Isolated execution.
- Trusted computing mechanisms can be used to:
  - Help mitigate threats to SDR; and
  - Reduce the impact of an attack in the event of threat realisation (defence in depth).

Application 2: DVB to a mobile

- Technological advances means that the potential exists to deliver complex content, e.g. broadcast video (DVB), to mobile consumers, with services available from many broadcasters.
- Trust is required between collaborating network operators and content providers.
- Content providers are increasingly aware of, and concerned about, copyright management,
- Current protection mechanisms are designed for relatively *static* receivers and services available from a small number of broadcasters.

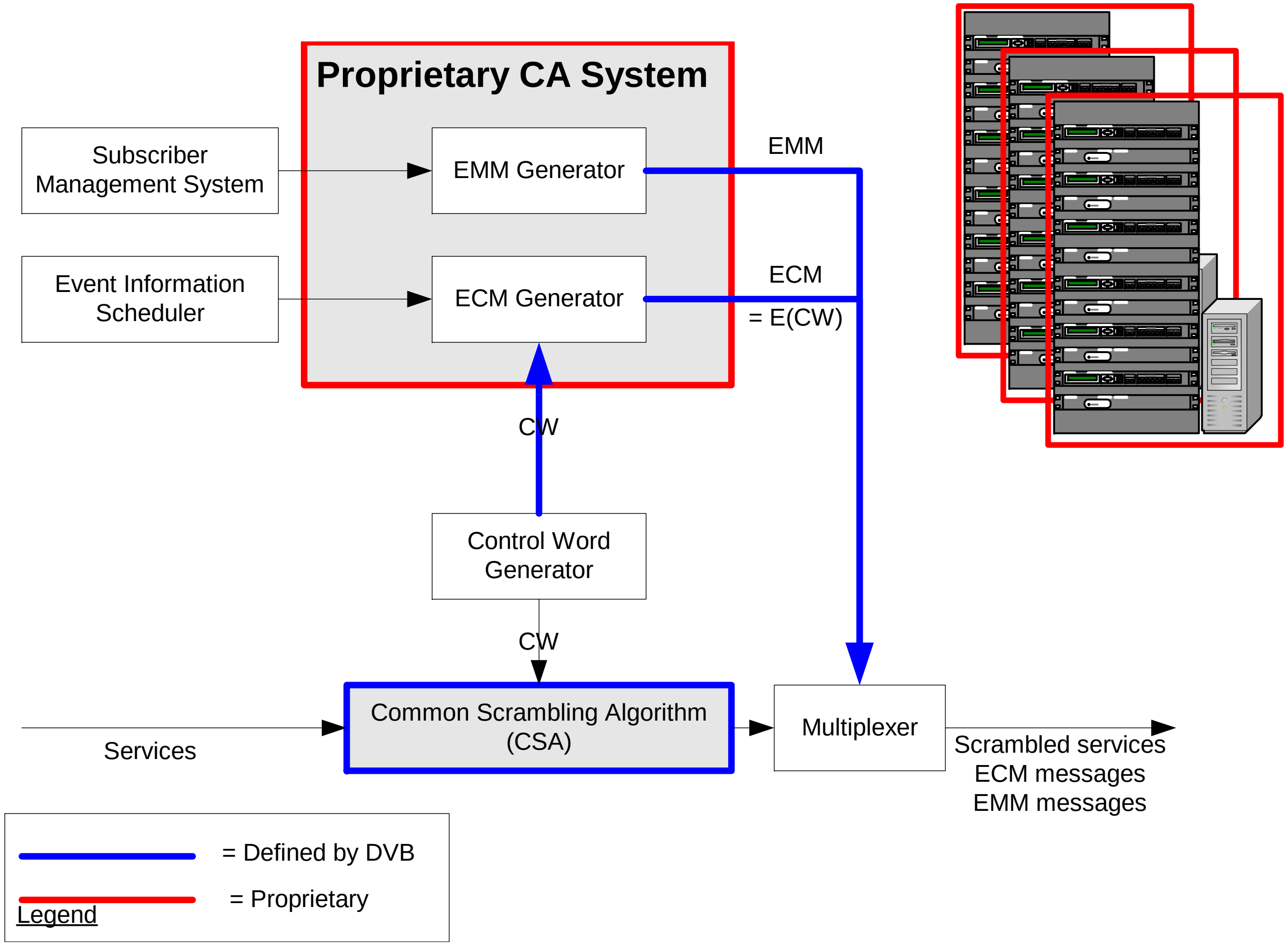
CA systems

- Broadcast content is currently protected by Conditional Access (CA) systems.
- These systems:
  - Scramble the video signal;
  - Manage keys and viewing rights;
  - Use proprietary security mechanisms within the framework of DVB standards;
  - The standards provide an interface to the proprietary systems.

DVB Standards

- Common Scrambling Algorithm: **ETSI ETR 289**
  - used to scramble and descramble services (video);
  - details available to all manufacturers.
- Simulcrypt: **ETSI TS 103 197**
  - supports multiple CA systems in parallel *at transmitter*;
  - uses common key to scramble services;
  - key encryption remains proprietary.
- Common Interface: **CENELEC 50221**
  - supports Common Interface Modules – PC Cards;
  - each card supports a single proprietary CA system *at receiver*.

Simulcrypt



Proprietary CA System

Subscriber Management System

Event Information Scheduler

EMM Generator

ECM Generator

EMM

ECM

= E(CW)

CW

Control Word Generator

CW

Services

Common Scrambling Algorithm (CSA)

Multiplexer

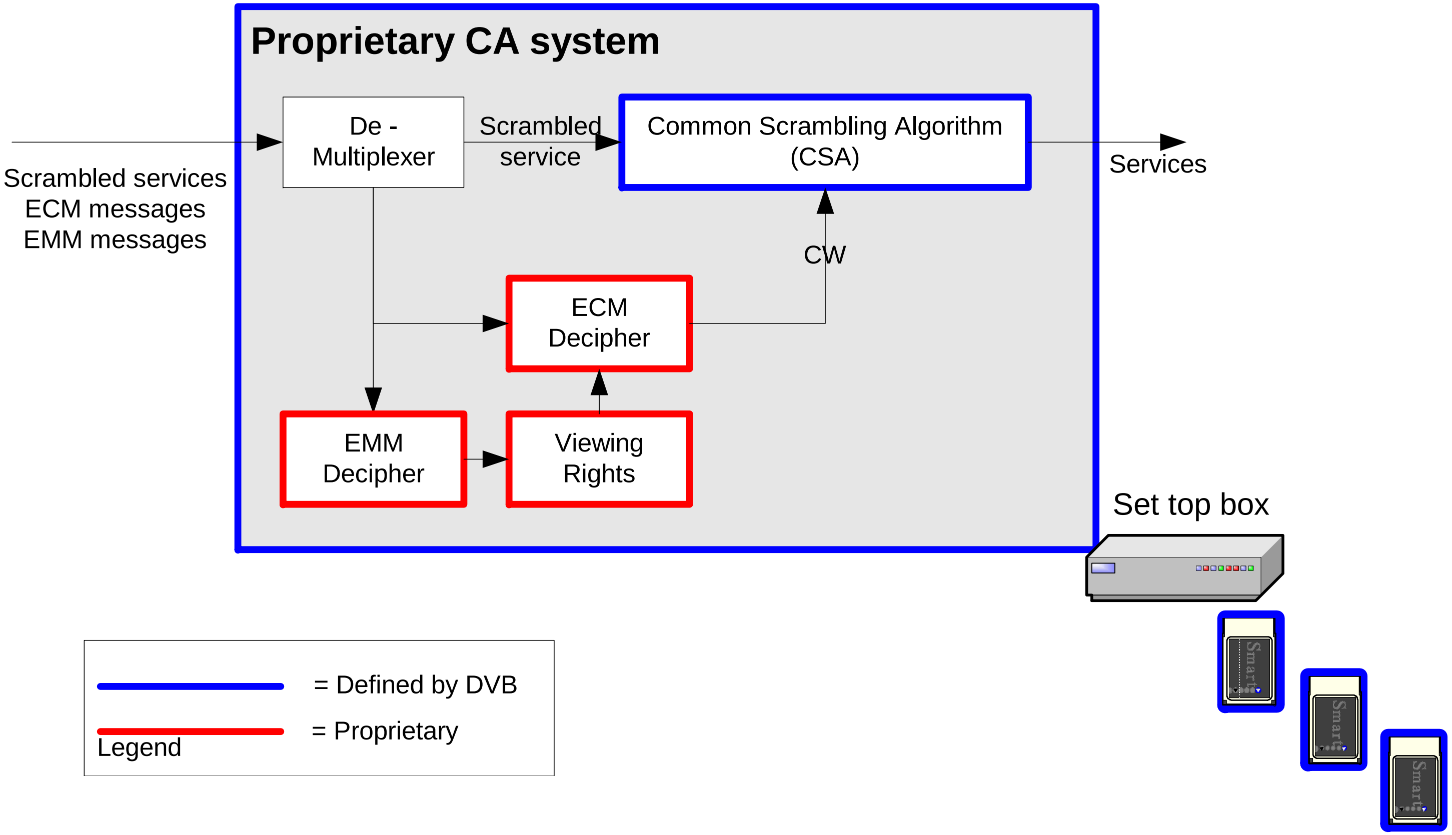
Scrambled services
ECM messages
EMM messages

Legend

= Defined by DVB

= Proprietary

Common Interface



Use of DVB standards

- DVB Standards:
  - provide a flexible interface to proprietary systems;
  - there are many proprietary systems.

CA systems – examples

- DVB-compliant Conditional Access (CA) systems include:

CA System	Vendor
Viaccess	Viaccess SA
NagraVision	Kudelski
Videoguard	NDS
Mediguard	Canal+
Mcrypt	Irdeto
PiSys	Irdeto
CryptoWorks	Philips
BetaCrypt	BetaResearch
Conax	Telenor

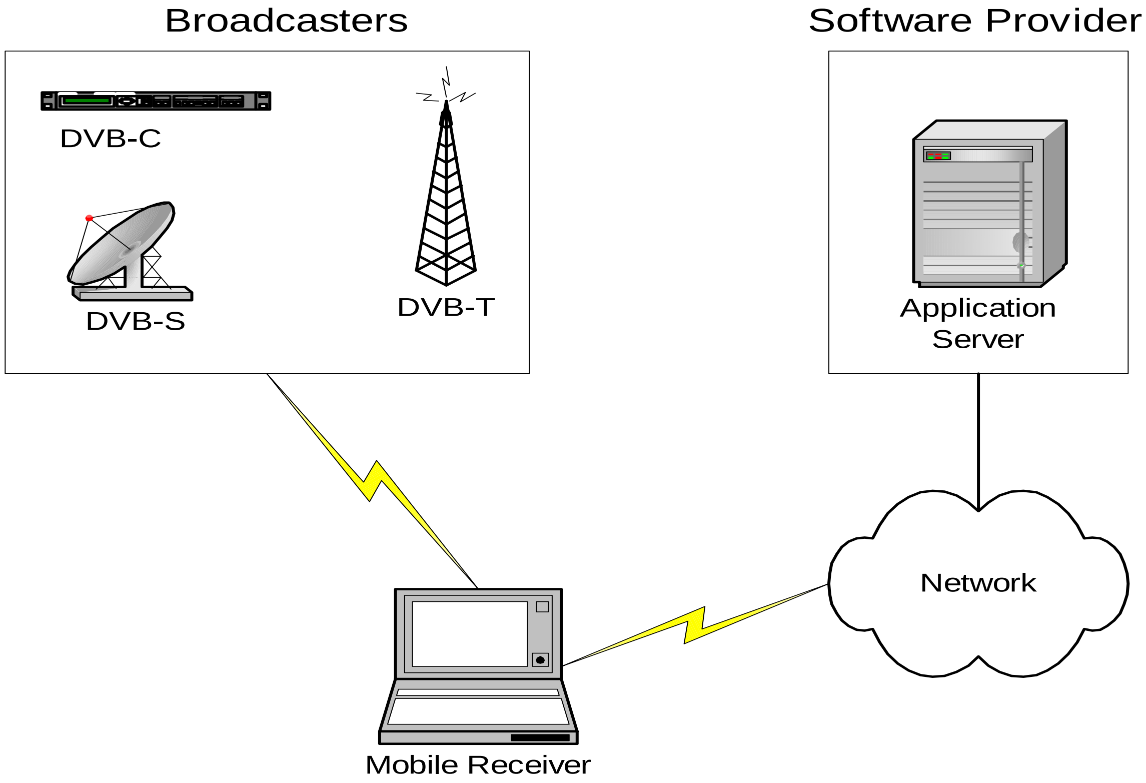
Limits of current approach

- The delivery of broadcast services to mobile receivers could involve many different broadcasters.
- Current protection mechanisms are designed for relatively static receivers and a small number of broadcasters.
- Common Interface:
  - Consumers require multiple PC-Card modules – high cost, inconvenience, lack of suitability for mobile devices.
- Simulcrypt:
  - Broadcasters need to install and maintain multiple CA systems, high cost for small ‘niche’ broadcasters.
- Current mechanisms not designed for mobile receivers

Potential Solution

- Download proprietary applications to mobile devices on demand.
- Problems:
  - applications and providers, are security sensitive;
  - trust required in the mobile host – threat of piracy, and hence protection needed for proprietary algorithms, keys;
  - receiving host needs to demonstrate that it can be trusted - application needs protection from the host.
- Trusted Computing provides mechanisms to demonstrate trust.

Model


Broadcasters
Software Provider
DVB-C
DVB-S
DVB-T
Application Server
Network
Mobile Receiver

Required security services

1. Confidentiality of application in transit.
2. Integrity of application in transit.
3. Entity authentication:
   - Host;
   - Application provider.
4. Origin authentication of application.
5. Freshness of messages.
6. Confidentiality and Integrity of application while in storage on the device (access control mechanisms to protect the application on the device).
7. Confidentiality and Integrity of application while executing on the device.

Corresponding security mechanisms

1. Symmetric encryption.
2. MACing of the application.
3. Entity authentication protocol runs as described later in this presentation;
   - Attestation (Host) as described within TCG TPM specification set.
4. Digital signature of the application provider on the symmetric keys used in 1 and 2.
5. Nonces/ timestamps.
6. Protected/secure storage, as described in TCG TPM v1.2 specification set.
7. Memory isolation techniques, as described by Microsoft with respect to NGSCB, for example.

Using TC for secure download

- We now consider general ways in which trusted computing technology might be used to support secure software download.
- The protocols we describe apply to both the applications we have outlined.
- Potentially also apply to many other scenarios requiring core software download to a mobile device.

Security requirements

- Demonstration of trustworthiness of receiving host:
  - Integrity challenge mechanism;
  - Integrity verification mechanism.
- Protection of downloaded software:
  - Secure delivery mechanism;
  - Secure execution environment.

Application of TCG technology

- Demonstration of trustworthiness:
  - – Integrity metrics:
    - authenticated boot – provided by CRTM;
    - configuration measurements – stored in PCR;
    - attestation – by TPM to current platform configuration.
- Software protection:
  - – Secure delivery mechanism – supported by key generation and exchange;
  - – Secure execution environment – sealed storage.

Other security and trusted platform technology

- Demonstration of trustworthiness:
  - Integrity *verification* mechanism – supported by certificates and Certification Authorities.
- Software protection:
  - Secure *delivery* mechanism – uses encryption and Message Authentication Codes;
  - Secure *execution* environment – builds on physical separation of trusted and untrusted processes:
    - Curtained memory – NGSCB, LaGrande;
    - Compartmentalised OS – e.g. NGSCB, Xen.

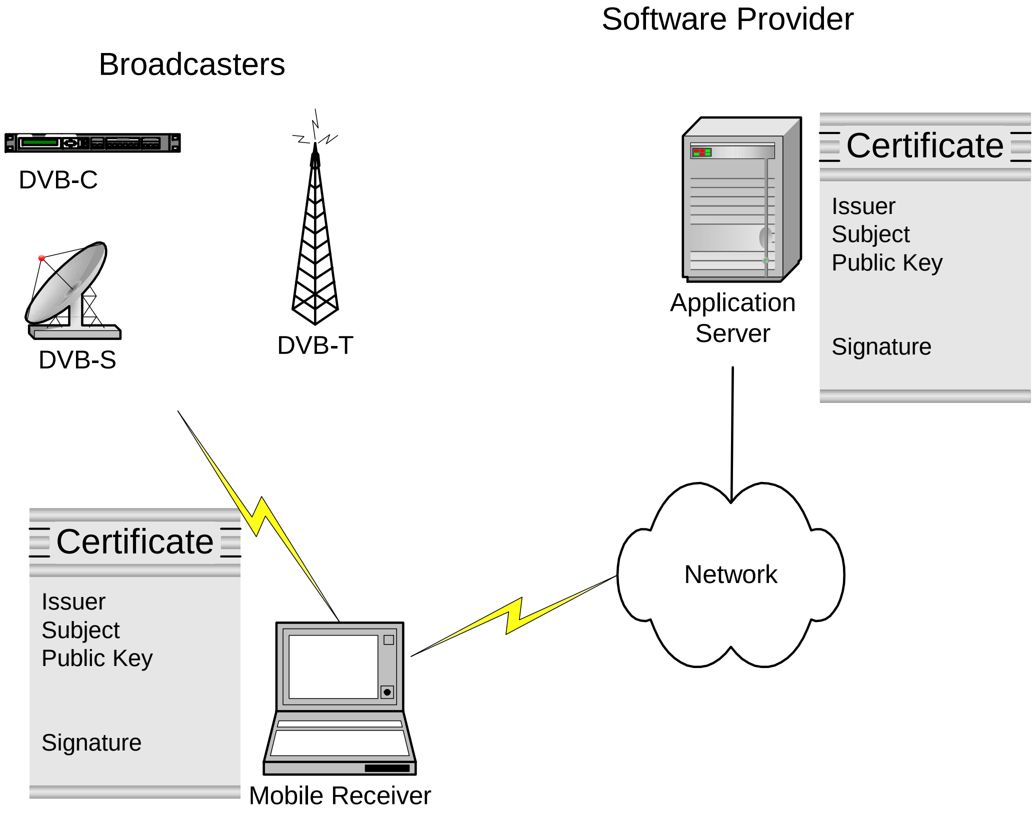
General approach to trusted download

- Demonstration of trustworthiness:
  - Authenticated boot;
  - Attestation of platform configuration;
  - Response to integrity challenge;
  - It is the challenger’s responsibility to verify the response and determine whether to trust the platform or not;
  - Host must not change configuration.
- Application protection:
  - Key exchange;
  - Keys in sealed storage to ensure consistent configuration;
  - Message Authentication Codes and encryption;
  - Isolation of applications.

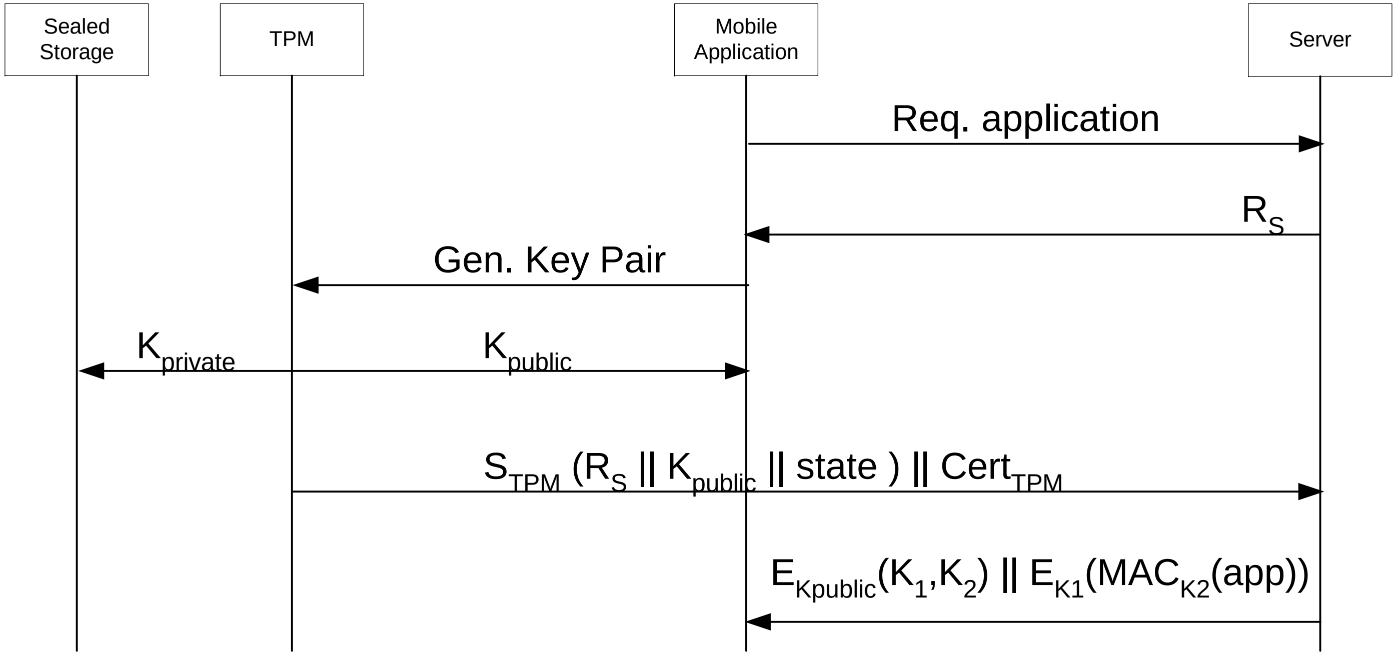
Protocol requirements

- The protocol must protect against:
  - **replay** – a malicious host could replay attestation information from before the system was compromised;
  - **tampering** – a malicious host could tamper with the integrity metrics before transmission to the challenger;
  - **masquerade** – a malicious host could replace the original integrity metrics with data from another system;
  - **revealing the application** – a malicious host could reveal the application and keys.

Model


Software Provider
Broadcasters
DVB-C
DVB-T
DVB-S
Certificate
Issuer
Subject
Public Key
Signature
Application Server
Network
Certificate
Issuer
Subject
Public Key
Signature
Mobile Receiver

Protocol 1

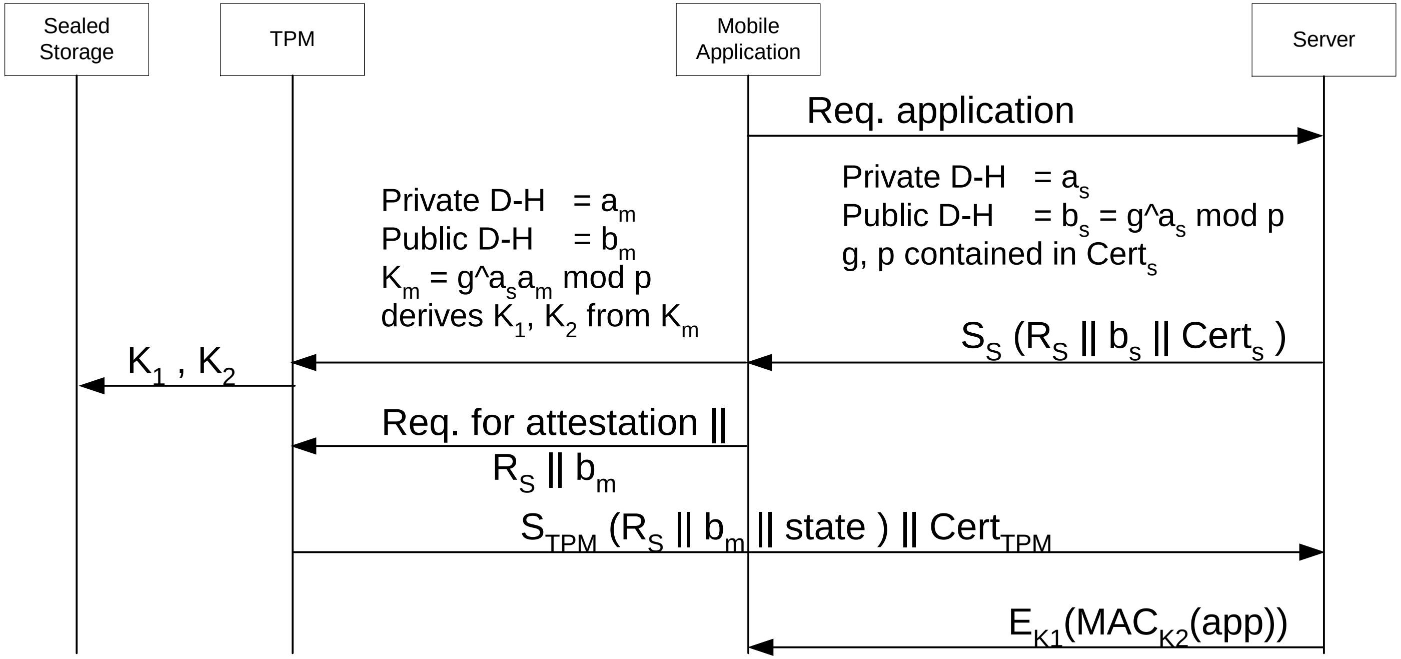


Sealed Storage
TPM
Mobile Application
Server
Req. application
R_S
Gen. Key Pair
$K_{private}$
K_{public}
$S_{TPM} (R_S \| K_{public} \| state) \| Cert_{TPM}$
$E_{Kpublic}(K_1,K_2) \| E_{K1}(MAC_{K2}(app))$

Protocol 1 properties

- The protocol protects against
  - Replay – the nonce, R_s, protects against replay;
  - Tampering – the TPM signature protects the integrity metrics;
  - Masquerade – the certificate of the TPM protects against masquerade;
  - Revealing the software – the keys K_1, K_2, protect the application during transmission; sealed storage and isolation protect during execution.

Protocol 2


Sealed Storage
TPM
Mobile Application
Server
Req. application
Private D-H = a_s
Public D-H = b_s = g^a_s mod p
g, p contained in $Cert_s$
Private D-H = a_m
Public D-H = b_m
K_m = g^$a_s a_m$ mod p
derives K_1, K_2 from K_m
S_S (R_S || b_s || $Cert_s$)
K_1 , K_2
Req. for attestation ||
R_S || b_m
S_{TPM} (R_S || b_m || state) || $Cert_{TPM}$
E_{K1}(MAC_{K2}(app))

Comparison of protocols

- The second protocol is more appropriate for resource limited devices:
  - it is less reliant on (costly) asymmetric encryption;
  - it makes fewer calls to the TPM.

Summary

- Using Trusted Platform technology:
  - the **Host** is able to demonstrate that it is running a secure execution environment;
  - the **Application provider** has confidence that software and data will not be accessed or modified;
  - the **User** has access to a wider range of applications

Contents

- TCG and the MPWG
- Implementing OMA DRM v2
- SIMLock using trusted computing
- Secure software download
- Concluding remarks

Resources

- http://www.isg.rhul.ac.uk/~cjm/tmp2.pdf [an article covering both talks]
- www.trustedcomputinggroup.org
- http://www.microsoft.com/windowsvista/default.aspx
- http://www.intel.com/technology/security/
- http://www.cl.cam.ac.uk/Research/SRG/netos/xen/
- http://os.inf.tu-dresden.de/L4/LinuxOnL4/
- http://www.opentc.net/
- Trusted Computing Platforms – TCPA Technology in Context, Siani Pearson (editor), HP Invent
- Trusted Computing – Chris Mitchell (editor), IEE

Acknowledgements

- Must thank all members of the OpenTC project, and, in particular, the RHUL team: Eimear Gallery and Stéphane Lo Presti.
- Must also thank Allan Tomlinson for material on SDR.
- Particular thanks to Eimear Gallery who produced the majority of the material for this talk.

Questions

- For further details on any topics addressed please contact me:
  - c.mitchell@rhul.ac.uk
  - http://www.isg.rhul.ac.uk/~cjm
  - Chris Mitchell
    Information Security Group
    Royal Holloway
    University of London
    Egham, Surrey TW20 0EX
    UK

The Open-TC project is co-financed by the EC.

If you need further information, please visit our website www.opentc.net or contact the coordinator:

Technikon Forschungs- und Planungsgesellschaft mbH
Richard-Wagner-Strasse 7, 9500 Villach, AUSTRIA
Tel. +43 4242 23355 – 0
Fax. +43 4242 23355 – 77
Email coordination@opentc.net

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.