

# Extending Trusted Computing as a Security Service

Chunhua Chen\*, Chris Mitchell and Shaohua Tang\*  
Information Security Group, Royal Holloway, University of London

## Abstract

We extend the Trusted Computing (TC) security infrastructure in a Generic Authentication Architecture (GAA)-like framework to enable the provision of security services, such as key establishment, to network applications.

## Background

### Generic Authentication Architecture:

- Standardised by 3GPP and 3GPP 2.
- A general framework that extends the cellular authentication infrastructure (includes UMTS and GSM) to enable the provision of security services to network applications.
- Consists of two procedures, GAA bootstrapping and use of bootstrapped keys.

### Trusted Computing (TC) Security Infrastructure:

A Trusted Platform (TP) compliant with the Trusted Computing Group (TCG) specifications is a computing platform with a tamper-proof and built-in Trusted Platform Module (TPM).

#### Properties of TPM:

- Protected capabilities, such as random number generation, asymmetric key generation, digital signing, encryption capabilities, etc.
- A unique Endorsement Key (EK) pair and a set of derived keys, such as an Attestation Identity Key (AIK).
- Other properties:

TPM, associated keys, protected capabilities, and the underlying Public Key Infrastructure (PKI) comprise a security infrastructure.



Figure 1: a trusted platform module.

Image source: [http://img.tomshardware.com/us/2008/02/11/how\\_hardware\\_based\\_security\\_protects\\_pcs/tpmchip.jpg](http://img.tomshardware.com/us/2008/02/11/how_hardware_based_security_protects_pcs/tpmchip.jpg)

## Core work

We make the TC security infrastructure play the role of the cellular authentication infrastructure in the GAA framework, and hence extend the TC security infrastructure to provide a security service, which we call TC GAA.

- Specify the architecture and components of TC GAA.
- Specify the interfaces and protocols between components.
  - Specify bootstrapping procedure of TC GAA, including an authenticated key agreement protocol.
  - Specify the derivation of an application-specific session key.
  - Specify use of bootstrapped key of TC GAA.

## Architecture Elements

### Bootstrapping Server Function (BSF):

- A new component, that acts as Trusted Third Party.
- Has a certified public key pair for entity authentication.

### Network Application Function (NAF):

- The server functionality of each GAA-aware network application.
- Assumed to have some means to set up a secure channel with BSF (e.g. as provided by SSL/TLS tunnel).

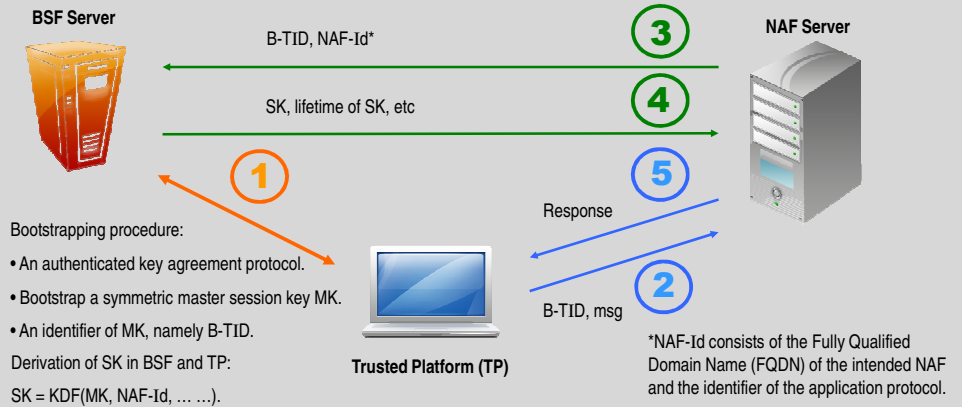
### TCG compliant Trusted platform (TP):

- The Endorsement Key for encryption.
- Has a certified public key pair for entity authentication (e.g. AIK).
- Protected capabilities as described in the protocol.

## Notation

- $Cert_x$ : A public key certification of entity X.
- MK : A symmetric master session key.
- SK : An application-specific symmetric session key.
- $R_x$  : A random number issued by entity X.
- $E_k(Z)$ : Encryption of data Z using the key K.
- H(Z) : A one-way hash function on data Z.
- $S_x(Z)$ : The digital signature of data Z computed using entity X's private signature transformation.
- X(public) : The public asymmetric key of X.
- X(private) : The private asymmetric of X.
- $Id_x$  : The identity of X.
- X||Y : The concatenation of data items X and Y in that order.
- $X \rightarrow Y : Z$ : Indicate that the message Z is sent by X to Y.

## Architectural Overview of TP-GAA



## Bootstrapping procedure

Bootstrapping procedure of TC GAA is used to bootstrap a new symmetric master session key between BSF and TP. It is an authenticated key agreement protocol specified as below:

- TP  $\rightarrow$  BSF: Request for bootstrapping master session key.
- BSF  $\rightarrow$  TP:  $R_{BSF}$ .
- TP: Generates a new temporary asymmetric encryption key pair (T(public) and T(private)) and certify the public key T(public) with an identity of T(public) chosen by TP user, namely,  $Id_{TP}$ .
- TP  $\rightarrow$  BSF:  $R_{BSF} || Id_{BSF} || T(public) || Id_{TP}$ .
- BSF: Retrieves  $Cert_{TP}$  and verifies it.
- BSF: Verifies  $S_{TP}(R_{BSF} || Id_{BSF} || T(public) || Id_{TP})$ .
- BSF: Verifies  $R_{BSF}$  to ensure the message is fresh and verifies that the message was intended for BSF.
- Assuming the signature from TP verifies correctly, the values of  $R_{BSF}$  and  $Id_{BSF}$  are expected, then BSF extracts T(public).
- BSF: Generates a symmetric key MK as master session key, and set lifetime of MK according to local policy. Generates an identifier B-TID of MK which consists of  $R_{BSF}$  and the domain name of BSF.
- BSF  $\rightarrow$  TP:  $E_T(public)(MK) || S_{BSF}(E_T(public)(MK))$
- TP: Retrieves  $Cert_{BSF}$  and verifies it.
- TP: Verifies  $S_{BSF}(E_T(public)(MK))$ .
- TP: Decrypts  $E_T(public)(MK)$  to get MK.

Steps 2 and 4 of the above protocol conform to the two pass unilateral authentication protocol described in clause 5.1.2 of ISO/IEC 9798-3:1998 where T(pub) serves as the nonce which is generated in every run.

The key agreement part of the protocol is a key transfer protocol.

After the procedure, BSF and TP share  $R_{BSF}$ ,  $Id_{TP}$ , B-TID, MK.

## Use of bootstrapped key

- TP: Derives an application-specific symmetric key SK as follow: SK = KDF(MK,  $R_{BSF}$ ,  $Id_{TP}$ , NAF-Id) where KDF is a key derivation function.  $Id_{TP}$  is the identity of T(pub), and NAF-Id consists of the Fully Qualified Domain Name (FQDN) of the intended NAF and the identifier of the application protocol.
- TP  $\rightarrow$  NAF: B-TID and msg. msg is the application request data secured using SK.
- NAF  $\rightarrow$  BSF: B-TID and NAF-ID. Note that it is assumed that a secure channel has been set up by some means between NAF and BSF.
- BSF: Derives SK = KDF(MK,  $R_{BSF}$ ,  $Id_{TP}$ , NAF-Id), and sets lifetime of SK according local policy.
- BSF  $\rightarrow$  NAF: SK, lifetime of SK, etc.
- NAF: Responses to the request using SK, if SK is valid.

## Contact information

### Name: Chunhua Chen

Address: Mcrea 106,  
Information Security Group,  
Royal Holloway, University of  
London, Egham, TW20 0EX  
Surrey, UK

T: 01784 443696  
E: [Chunhua.chen@rhul.ac.uk](mailto:Chunhua.chen@rhul.ac.uk)

Chunhua Chen is a Phd student under supervision of Prof. Shaohua Tang in School of Computer Science and Engineering, South China University of Technology. This work was performed during a visit to the Information Security Group at Royal Holloway, University of London, sponsored by the Chinese Scholarship Council.