

# Using existing security infrastructures

Chris Mitchell

Information Security Group  
Royal Holloway, University of London  
<http://www.isg.rhul.ac.uk/~cjm>

# Acknowledgements

- This is joint work with Chunhua Chen and Shaohua Tang (South China University of Technology).
- Chunhua Chen is currently visiting RHUL.

# Contents

- Security infrastructures
- GAA
- UMTS-GAA
- TC-GAA
- Applying GAA variants
- Conclusions

# Security infrastructures

- In order to use cryptography to protect communications, some kind of security infrastructure needs to be in place.
- In its simplest form, this just means setting up shared secret keys between communicating parties.
- Traditionally, e.g. in banking networks, this can be achieved using one or more Trusted Third Parties (TTPs).
- One type of TTP for this purpose is known as a Key Distribution Centre (KDC).
- A KDC shares a secret key with every party, and these keys can be leveraged (using an appropriate protocol) to set up a secret key between any two parties.

# Public Key Infrastructures (PKIs)

- A PKI is another type of security infrastructure, based on digital signatures.
- A Certification Authority (CA) creates digitally signed certificates for user public keys, binding a user name to a public key.
- Anyone with a trusted copy of the CA public key can verify the certificates is has generated, and thereby obtain ‘trusted’ public keys.

# The promise of a universal PKI

- A few years ago, PKI was the subject of huge hype.
- Companies producing PKI products (e.g. CA software) or providing PKI services suddenly (and temporarily!) became hugely valuable.
- In many cases the vision sold as part of this hype was of some kind of universal PKI, whereby every PC in the world would have a public key certificate, which could then be used for a huge range of purposes, e.g.:
  - secure e-commerce;
  - universal secure e-government;
  - secure home banking;
  - electronic signatures for all; ...

# PKI – what happens in practice I

- Of course, this has not happened.
- There are many PKIs, each set up for a specific purpose.
- For example:
  - companies have their own PKIs, used to support internal secure communications;
  - MasterCard and Visa (and card issuing banks) have PKIs set up to support EMV (used to support smart card based credit/debit card transactions, e.g. in parts of Europe);
  - Internet web sites have certificates used for SSL/TLS security.
- There are, of course, many explanations for this – one being the fact that the policies under which certificates are issued will depend on the context of use.

# PKI – what happens in practice II

- More generally, PC users do not have the expertise or motivation to generate a signature key pair, and obtain a certificate for their public key.
- This can be seen from the failure of the SET e-commerce secure payment system, one of the major obstacles to the adoption of which was the need for every user to generate a key pair, and take a copy of their public key to their bank.
- End users cannot be expected to understand the operation of public key cryptography.
- Moreover, current PCs typically do not have a means for secure key storage (needed for the private key).



# Re-using infrastructures

- We do not have a universal security infrastructure (yet, anyway).
- Setting up a new infrastructure for every application is very costly.
- Hence, can we use existing security special-purpose infrastructures to provide a ‘cut price’ universal security infrastructure?
- This is the subject of this talk ...

# Contents

- Security infrastructures
- GAA
- UMTS-GAA
- TC-GAA
- Applying GAA variants
- Conclusions

# Background

- The term *Generic Authentication Architecture* (GAA) has been developed within the mobile phone community.
- It refers to a standardised way of exploiting the mobile phone security infrastructure to provide general purpose authentication and key management services.
- The mobile operator (or an entity with a close relationship with an operator) acts as a TTP.
- We start by describing this architecture in general terms.

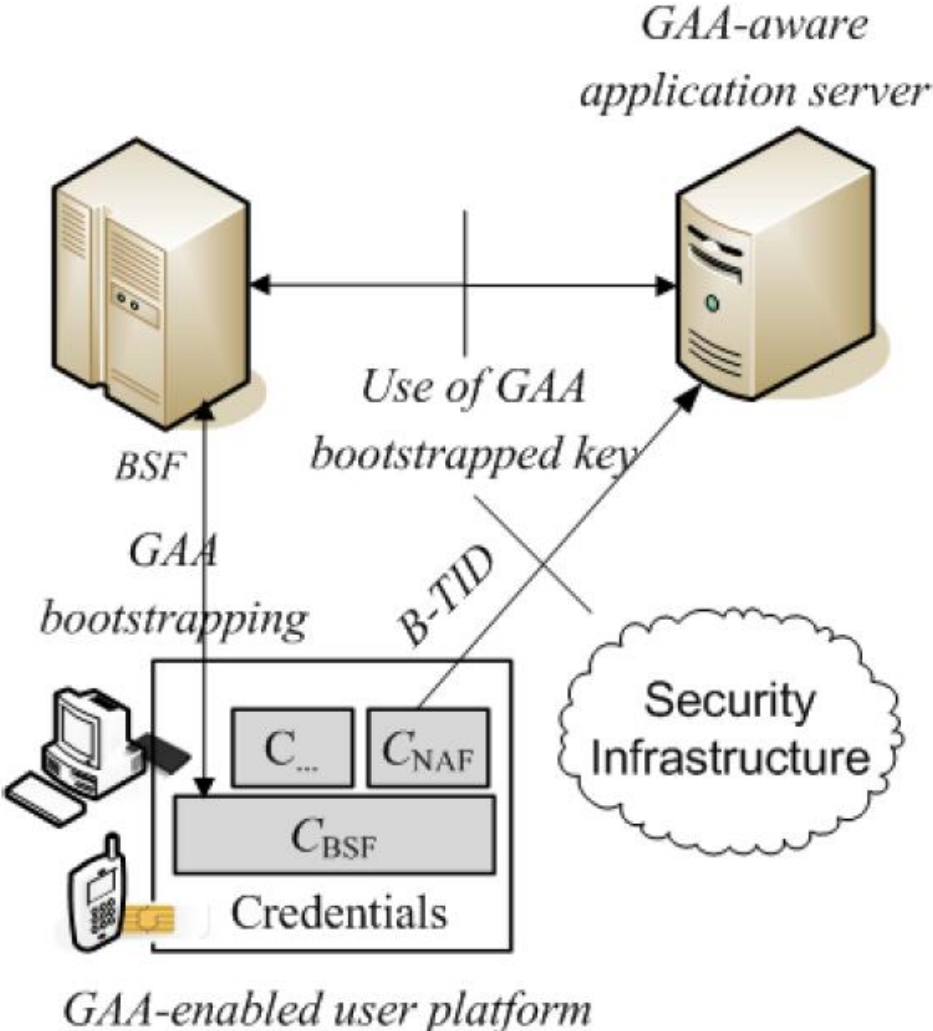
# GAA roles

- The GAA architecture involves three roles:
  - **Bootstrapping Server Function (BSF)** – this is the TTP that provides the service;
  - **GAA-aware application server** – has a trust relationship with the BSF;
  - **GAA-enabled user platform** – has an existing security relationship (e.g. shared secret key) with the BSF.

# GAA service

- GAA establishes an authenticated **application- and server-specific** secret key between the *GAA-enabled user platform* and an arbitrary *GAA-aware application server*.
- The user (with a GAA-enabled user platform) must have a mobile phone subscription.
- The target server must have a relationship with the GAA service provider.

# GAA overview



# GAA procedures

- Two main procedures:
  - **GAA bootstrapping** – Establishes a secret master key  $MK$  (and an identifier  $B-TID$  for the key and a key lifetime) between GAA-enabled user platform and the BSF.
  - **Use of bootstrapped keys** – Establishes an application- and server-specific session key  $SK$  between platform and server using  $MK$  [ $MK$  is not divulged to the server]:

$$SK = f(MK, \text{server-ID}, \text{app-ID}, \dots)$$

where  $f$  is a key derivation function.

# Key provisioning

- The GAA-enabled user device can calculate  $SK$  for itself.
- The GAA-enabled server is provided with  $SK$  by the BSF.
- Thus a secure channel between the BSF and the server is required.



# Our goal

- GAA was designed specifically for use with the 3G mobile telecommunications security infrastructure (we call this UMTS-GAA).
- We show how to provide GAA-like services with other pre-existing infrastructures.
- As a result, any services built on UMTS-GAA can immediately be migrated to other security infrastructures.

# Contents

- Security infrastructures
- GAA
- UMTS-GAA
- TC-GAA
- Applying GAA variants
- Conclusions

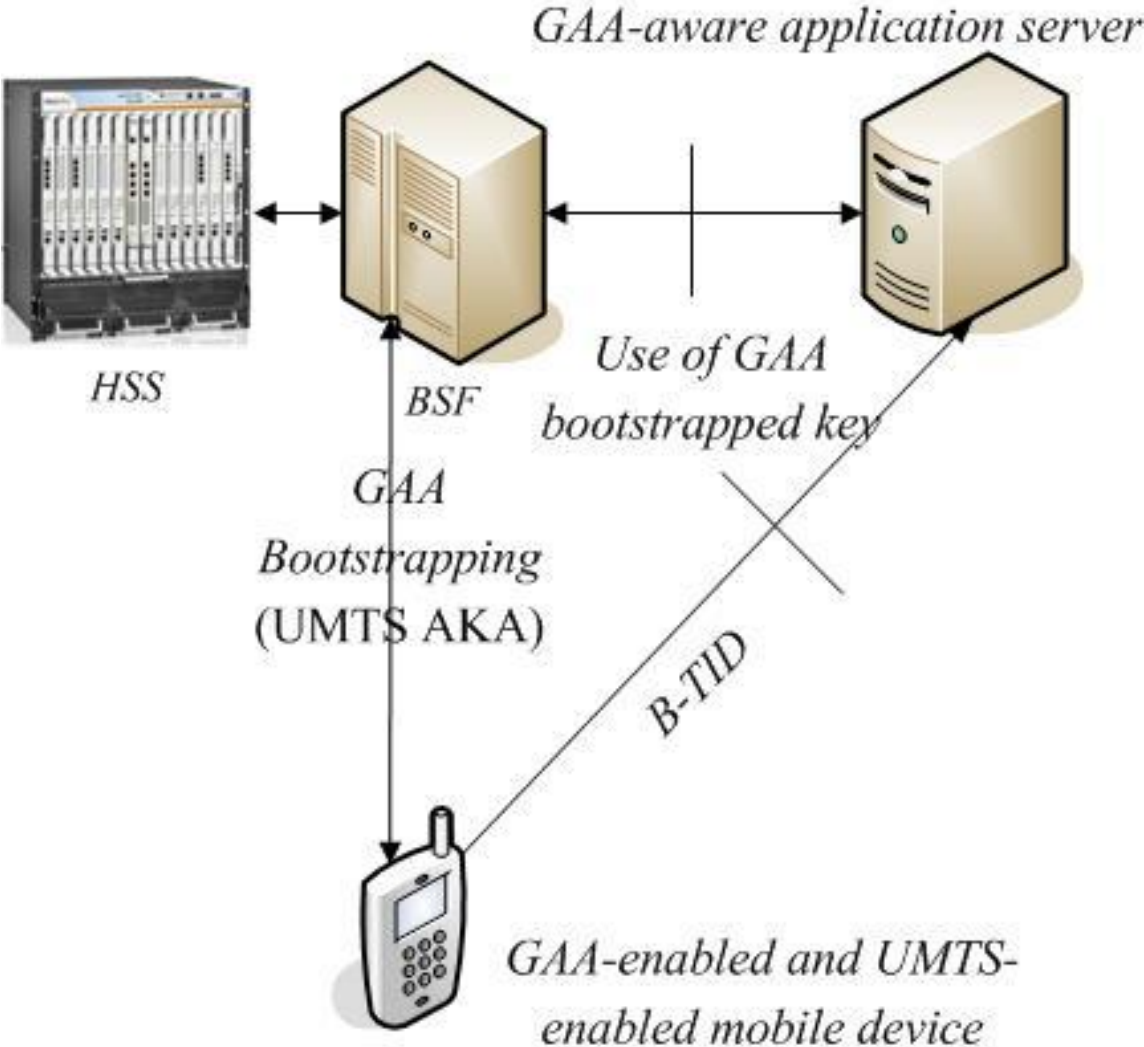
# UMTS – background

- The UMTS security infrastructure (supporting mobile phone security) has the following roles:
  - **USIM** – smart card held by user (in phone);
  - **Home Subscriber Server (HSS)** – shares secret key with USIM, and operated by mobile phone service provider with whom user has contractual relationship.

# UMTS-GAA

- In UMTS-GAA:
  - GAA-enabled user platform is a UMTS-enabled mobile device, with a USIM;
  - BSF connects to the appropriate HSS for the USIM (may be owned by same operator);
  - UMTS Authentication and Key Agreement protocol (UMTS AKA) is used to establish *MK* between GAA-enabled user platform and BSF (*MK* is concatenation of *IK* and *CK*).

# UMTS-GAA



# Session key derivation

- In use of bootstrapped keys:

$$SK=f(MK, RAND, mobile-ID, server-ID, app-ID, \dots)$$

- RAND is the value used in the UMTS AKA protocol (functions as a random challenge in the protocol).

# Contents

- Security infrastructures
- GAA
- UMTS-GAA
- TC-GAA
- Applying GAA variants
- Conclusions

# TC – a universal security infrastructure?

- Trusted computing provides another security infrastructure.
- Every PC owner will have a crypto-capable PC, with an asymmetric key pair in their TPM and a public key certificate for the public key.
- Moreover, the TPM is capable of generating signature key pairs on demand, of using generated private keys to sign arbitrary data, and of providing secure storage for private keys.



# Possible problems

- The key pair provided in every TPM (when shipped to user) is not suitable for use as a general purpose key pair:
  - although it is an RSA key pair, it is intended for use as an encryption/decryption key pair;
  - the TPM does not enable its use for signing arbitrary data.
- The TPM is capable of generating an RSA key pair designed for signing (known as an AIK – Attestation Identity Key), and can also obtain an X.509 certificate for the public part of the AIK from an entity known as a Privacy-CA.
  - However, the private part of the AIK cannot be used to sign arbitrary data.

# Solutions to problems

- Get the TPM to generate another signature key pair, and use an AIK to sign a 'certificate' for the public key.
- The private key of this key pair can be used to sign arbitrary data.
- This means that the PC now has a means of generating arbitrary numbers of signature key pairs (essentially automatically) and obtaining certificates for them.
- Only problems are:
  - There is a need to associate two certificates with each key pair (the Privacy-CA certificate for the public AIK, and the AIK-signed certificate for the public key in use);
  - The AIK-signed certificate is not in the standard (X.509) format.

# Certificate 'translation'

- A means of addressing these last two problems has been proposed by the TCG.
- Proposed special extension to PKCS#10 (PKCS#10 is a format for submitting certification requests to a CA).
- This extension (SKAE) allows a PC to submit a PC-generated certificate (signed using AIK) for signature public key, with other evidence, as part of a cert request.
- CA verifies the certificate and evidence, and would then generate a new certificate for the PC public key.
- All these processes could be performed by a Java applet, which would give the PC user a secure and automatic means of joining a global PKI.

## Example 1 : SSL client side authentication

- Currently, SSL is only used for unilateral authentication i.e. of the server to the client, mainly because client PCs typically do not have key pairs and certificates.
- Precisely the procedure just described could give a means for a PC user to acquire a signature key pair and a public key certificate in order to support SSL client side authentication.
- This is described in greater detail in:
  - A. Alsaïd and C. J. Mitchell, 'Preventing phishing attacks using trusted computing technology', in Proc. INC 2006, 6th International Network Conf., Plymouth, July 2006, pp.221-228.
- Related work, including implementations, has been conducted by the OpenTC project.

## Example 2: Secure PC-based electronic signatures

- A considerable amount of work has gone into developing legislative and commercial frameworks for electronic signatures.
- However, such frameworks typically require a cumbersome registration procedure for users, and also some means of storing private keys securely.
- The possibility exists that, with the aid of the TPM in a PC, the PC itself can become a trusted platform for the implementation of a personal electronic signature capability, since it can provide the secure storage and also automatically perform the registration procedures.

# Portability and privacy issues

- The problem remains that PCs are not typically in one-to-one correspondence with human users.
- Users have multiple PCs (transferring secrets between TPMs is difficult), and PCs may have multiple users.
- In the latter case, issues may arise in holding a single user accountable for the behaviour of a PC.
- However, TPMs are ‘owned’ by a single user, which typically means that only one individual will be able to use the TPM-protected keys.
- If users want multiple ‘unlinkable’ identities, TPM can generate new key pairs. (Privacy-preserving certification and use of cryptography is key feature of TCG specs.).

# Using the TC infrastructure directly

- It is perfectly possible to design applications building directly on the trusted computing infrastructure.
- Substantial literature now exists.
- However, secure application protocols are non-trivial to design.
- Trust relationships can be very unclear.

# Third party support

- We propose the creation of a GAA-like third party based service to enable the provision of security services building on the TC infrastructure.
- The definition of standard security services, e.g. for key establishment, will enable the TC infrastructure to be exploited in a simple and uniform way.
- It will also provide an opportunity for trusted computing aware third parties to provide novel security services.

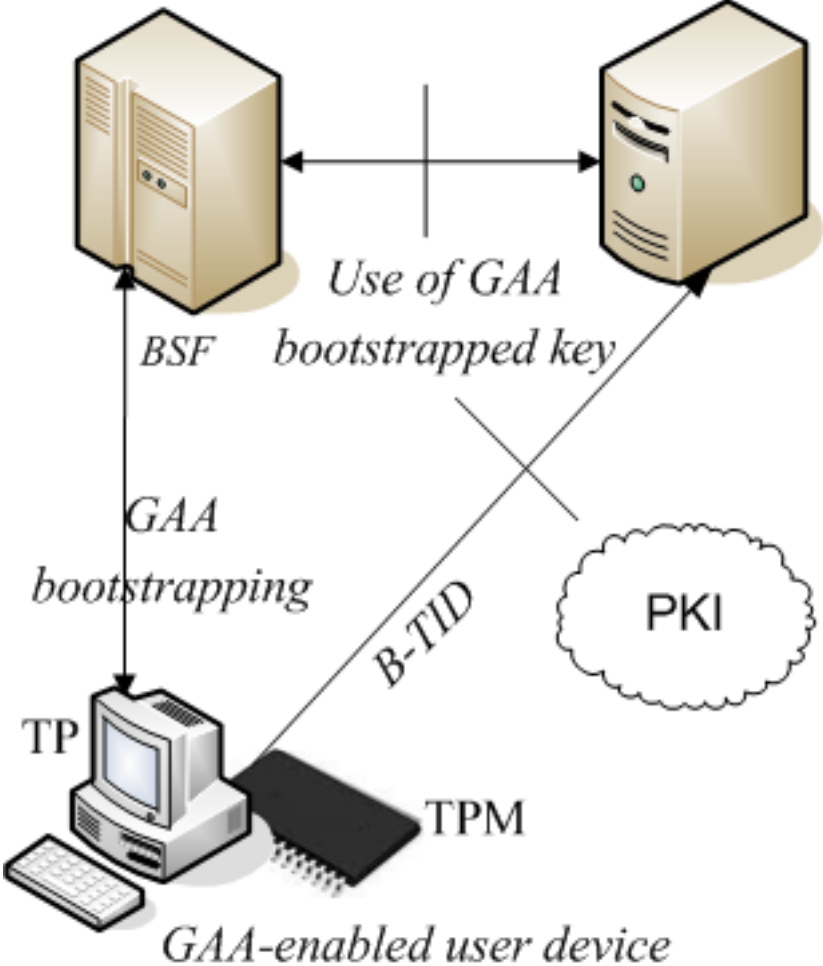


# Using the GAA architecture

- We have designed a version of GAA (which we call **TC-GAA**) which enables TC to be used to provide generic security services in a simple and uniform way.

# TC-GAA

*GAA-aware application server*



# TC-GAA – a sketch

- The TPM on the client machine is instructed to generate a new encryption key pair.
- The public key is then signed (certified) by the TPM using a previously generated Attestation Identity Key (AIK).
- The newly generated certificate is now sent to the BSF along with a previously generated Privacy-CA-generated certificate for the AIK public key.
- After verifying the two certificates, the BSF generates an *MK*, encrypts it using the TPM-generated public key, and ships it back to the TPM.
- This complete the TC-GAA bootstrapping procedure.

# TC-GAA properties

- Note that the derivation of  $SK$  can be very similar to the generic case.
- It is interesting to observe that, unlike UMTS-GAA, the ‘issuer’ of the TPM is not actively involved.
- Any TTP can function as the BSF without a trust relationship with a further third party.
- This enhances the privacy properties.
- This advantage results from building GAA on asymmetric crypto rather than shared secrets.

# GAA as a general framework

- GAA was originally designed to provide a way of exploiting the mobile phone security infrastructure.
- We have shown how it can be used to build on the TC infrastructure.
- Could also be used as a framework for providing general purpose security services building on other pre-existing security infrastructures.

# EMV-GAA (sketch)

- A further existing security infrastructure which could be used as the basis of a GAA service is the ‘chip and PIN’ infrastructure.
- Every EMV card shares a (unique) secret key with the card issuing bank.
- This suggests something very similar to UMTS-GAA could be designed, with the card issuer taking the role of the HSS.
- The user would need a card reader and an appropriate PC application.

# EMV-GAA – further developments

- Some EMV cards (supporting CDA or DDA as opposed to the widely used SDA) possess an RSA key pair and a certificate chain for the public key.
- Such a card can be requested to compute a signature by any card reader.
- This could be used to support a more TC-like type of GAA.
- It could also function as the basis of something like a universal PKI.

# Contents

- Security infrastructures
- GAA
- UMTS-GAA
- TC-GAA
- Applying GAA variants
- Conclusions



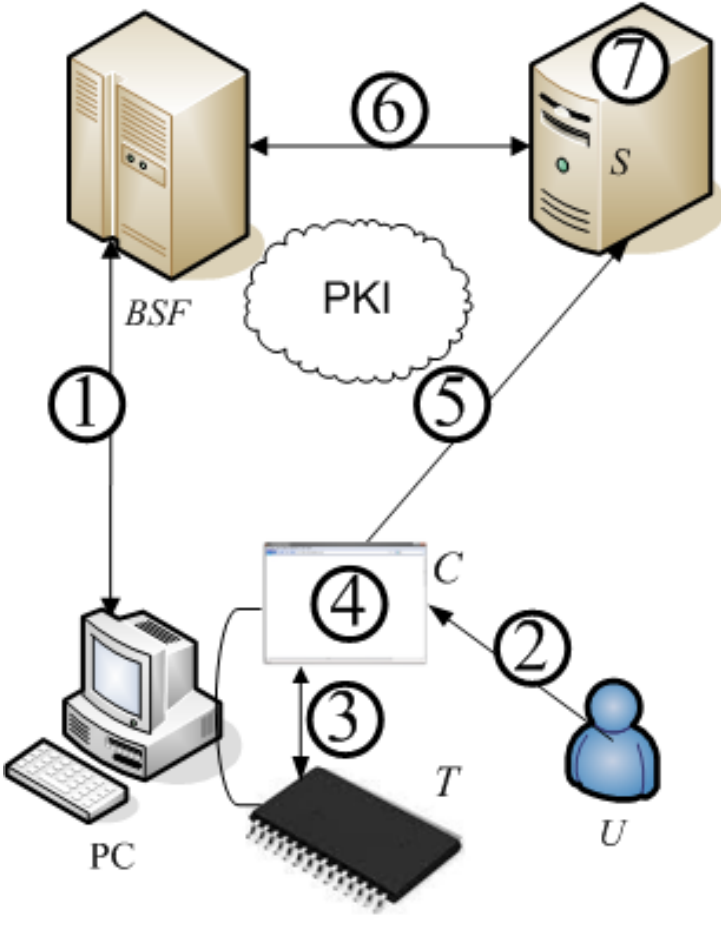
# GAA-based one-time passwords I

- We consider one possible application of TC-GAA, namely to enable the simple derivation of one-time passwords (OTPs).
- These passwords are based on a (potentially weak) long-term user password.
- The TC-GAA session key provides protection against brute force password searches.

# GAA-based one-time passwords II

- The OTP is computed as a function of the long-term user password and the short term application-specific session key.
- Compromise of the OTP does not enable a brute-force search for the password without knowledge of the session key.
- The TP used in the protocol does not need to be registered to the user – only needs to be trusted not to compromise the password.

# TC-GAA-OTP



- ①  $T \leftrightarrow BSF$ : bootstrap GAA credentials.
- ②  $U \rightarrow C$ : username and pw.
- ③  $C \leftrightarrow T$ : interact for a session key  $SK$ .
- ④  $C$ : computes  $otp = f(SK, pw)$ .
- ⑤  $U(C) \rightarrow S$ :  $B-TID$ , username and  $otp$ .
- ⑥  $S \leftrightarrow BSF$ :  $S$  fetches  $SK$  and its lifetime.
- ⑦  $S$ : checks whether or not  $SK$  is valid; if so,  $S$  recomputes  $otp$  for authentication; if not,  $S$  discards the request.

# GAA OTP – other instantiations

- The notion of using a GAA session key to help generate an OTP from a long-term weak password applies to all instantiations of GAA.
- Indeed, in parallel work we have designed a series of simple OTP schemes using a GAA-enabled mobile phone.

# GAA-based SSO

- We are also developing ways in which GAA could be used to build more general identity management solutions, including single sign-on schemes.
- Some work along these lines has already been standardised for UMTS-GAA, notably interoperation with CardSpace, OpenID and Liberty.

# Contents

- Security infrastructures
- GAA
- UMTS-GAA
- TC-GAA
- Applying GAA variants
- Conclusions

# Building the TC infrastructure

- There is a major problem with rolling out trusted computing applications.
- The envisaged complex infrastructure does not yet exist.
- TC-GAA may help with providing the business case necessary for the emergence of the wide range of third party security services necessary to fully realise the goals of trusted computing.

# TPM.next

- The TC-GAA scheme we have proposed is built on the current generation of TPM (v1.2) functionality.
- A new set of TPM specifications (with working name TPM.next) is due to be released shortly.
- Whilst backwards compatible, these allow a richer range of functions, and may make certain tasks simpler.



# Trust

- All these GAA-based schemes require some level of trust in the TTP providing the BSF functionality.
- The exact degree of trust depends on the application.
- This may be a problem for some applications, but not for others, particularly for corporate environments.
- In any case, we all depend on TTPs for a variety of aspects of daily life (including banking, telephony, shopping, ...).

# Questions ...