GAA/GBA: a new Architecture for single sign-on

2nd ETSI Security Workshop: Future Security
16-17 January 2007 Sophia- Antipolis (France)

Wednesday, 17th January 2007
luisangel.galindosanchez@telefonica.es
Agenda

01 GAA
02 EXAMPLE OF GAA APPLICATION
03 GBA
04 SSC
05 HTTPS
06 CONCLUSIONS
Agenda

01 GAA
02 EXAMPLE OF GAA APPLICATION
03 GBA
04 SSC
05 HTTPS
06 CONCLUSIONS
GAA (Generic Authentication Architecture)
# GAA Protocols

<table>
<thead>
<tr>
<th>GAA SSC TS 33.221</th>
<th>GAA HTTPS TS 33.222</th>
<th>GAA System Description TR 33.919</th>
<th>GAA GBA TS 33.220</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ua</td>
<td>Ub</td>
<td>Zh</td>
<td>Zn</td>
</tr>
</tbody>
</table>

- **HTTP Digest RFC 2617**: Supports potentially many protocols on Ua Interface (e.g., HTTP Digest)
- **HTTP Digest AKA RFC 3310**:
- **IMS Cx Diameter message definitions TS 29.229**:
- **HTTP**
- **Diameter RFC 3588**:
- **TCP**
- **SCTP**
- **IP**

*The GAA supports potentially many protocols on Ua Interface (e.g. HTTP digest)*
01 Basic Authentication Mechanism

- GBA (Generic Bootstrapping Architecture)
  - Mechanism of application independent based on 3GPP AKA
  - Shared key between client and AS
  - Eg.: HTTP Digest, IKE …

- SSC (Support for Subscriber Certificates)
  - Authentication based on public-private key and certificates
  - A PKI infrastructure is needed
  - Eg: PGP, HTTP over TLS

- Access to AS by HTTPS
  - User authentication based on HTTP over TLS
  - Possible with GBA and SSC
Agenda

01 GAA

02 EXAMPLE OF GAA APPLICATION

03 GBA

04 SSC

05 HTTPS

06 CONCLUSIONS
02 General Description

- Example: Buy contents in a content provider with secure commercial transactions
  - 1. Get certificate for authenticating with content provider (first time)
  - 2. Authentication is done

- Steps
  - 1. Authentication for a secure context establishment based on shared key (GBA)
  - 2. Use the secure context to get certificates from PKI (SSC)
  - 3. Use of certificates with third parties (SSC)

GAA (Generic Authentication Architecture)
02  Entities

- 1. UE - BSF (Ub): Negotiation of session shared key Ks (GBA)
- 2. UE – Portal PKI (Ua): Process for getting certificates for user and CA (SSC)
- 3. UE – AS (Ua): User uses his/her certificate with third parties based on PKI (SSC)
A security association between UE and PKI is created (confidentiality, integrity...)

Key generation:
1. UE sends HTTP request initial
2. HSS sends 401 Unauthorized
3. UE sends HTTP request
4. BSF sends Get AV
5. UE sends 401 Unauthorized (challenge)
6. BSF sends Challenge response
7. UE sends 200 OK

Ks = Ck || Ik
02 User’s Certificate

- UE
- PKI
- BSF

8- Challenge response
9- Gets Ks_NAF
10- 200 OK

 Generates Ks_NAF from Ks and B_TID

Authenticates and generates certificate

Verify values of Authentication-Info header. If success, certificates is accepted
CA’s certificate

11- Request CA’s certificate

12- 401 Unauthorized

13- Response challenge

If UE has no keys, they are generated by bootstraping process

Autenticates and generates certificate. Access to BSF for getting info needed

14- 200 OK

UE receives the response, checks values and if success accepts certificate
02 Access to the service provider

- User has user’s certificate with public key signed with CA’s private key
- User has CA’s certificate with CA’s public key
- User has the following security services
  - Authentication
  - Integrity
  - Confidentiality
  - No repudiation
Agenda

01 GAA
02 EXAMPLE OF GAA APPLICATION
03 GBA
04 SSC
05 HTTPS
06 CONCLUSIONS
03 GBA Architecture

- **Zn-Proxy (Diameter Proxy)**
  - Proxy between visited NAF and local BS
  - Creates a secure connection and checks if there is communications between NAF and BSF
  - Security level ≥ highest NAF’s security level
  - If BSF and NAF are in different networks, Zn and Zn’ shall be protected by TLS

- **BSF (Bootstrapping Server Function)**
  - Controlled by local operator
  - In charge of establishing shared key with UE using AKA

- **NAF (Network Application Function)**
  - Similar to an AS in IMS

- **HSS (Home Subscriber Server)**
03 GBA Operations

- Mutual authentication UE and BSF based on AKA
- After this **sessions keys are agreed**

- BSF access to HSS in order to get info for GBA process
  - GUUS (**GBA User Security Settings**)

- After bootstrapping between UE and BSF, UE and NAF may use some authentication protocol based on key sessions previously generated
  - **Ks_NAF key generated by Ks** will be used for secure connections
03 GBA Push extension

- Early stage of development
- No need for UE or BSF to start bootstrapping
- All bootstrapping goes through NAF
- ‘Up’ defined to transmit Push information from BSF to UE
03  2G GBA extension

- Problem:
  - MILLIONS of users still with SIM cards!
  - Security in 2G lower than in 3G

- Objective
  - Enables GAA services for 2G users ⇒ adapt GAA elements for using with 2G GBA

- Main changes
  - *Bootstrapping* remain without significant changes
  - Parameter and algorithms adaptation in order to use with 2G GBA
Agenda

01 GAA
02 EXAMPLE OF GAA APPLICATION
03 GBA
04 SSC
05 HTTPS
06 CONCLUSIONS
04 Characteristics

- Based on public/private key mechanisms and digital certificates
- PKI infrastructure is needed
- More security than shared key
- User gets certificates from CA and access to AS with them
- Works in roaming
- Home operator controls subscriber's certificates emission based on GUSS (GBA User Security Settings)
- Uses X.509 certificates
**04 SSC Architecture**

BSF provides the authentication info needed by PKI
This info is in USS (User Security Settings) stored in HSS

PKI acts as a CA and RA
Request and responses to/from PKI shall be protected (Ua interface) with shared key previously generated between UE and BSF

UE shall be able to generate pairs of public/private keys, to store the private key and to protect the private key
Agenda

01  GAA
02  EXAMPLE OF GAA APPLICATION
03  GBA
04  SSC
05  HTTPS
06  CONCLUSIONS
Access to AS by HTTPS

- Authentication between UE and AS using HTTPS
  - Without *Authentication Proxy*
    - AS authenticates directly the UE using GAA mechanisms
  - *Authentication Proxy (NAF=AP)*
    - Authorization and authentication delegated to AP
    - AP authenticates the UE and resend the UE’s identity to AS
    - AP can use any GAA mechanism for UE authentication

![Diagram of UE, AP, and services](image)
Agenda

01  GAA
02  EXAMPLE OF GAA APPLICATION
03  GBA
04  SSC
05  HTTPS
06  CONCLUSIONS
## CONCLUSIONS

- **GAA**
  - Created in order to ensure an security environment in accessing to services in AS from a mobile handset
  - 3GPP standards are based in PKI and symmetric technologies
  - Two main mechanism:
    - Shared key: GBA.
    - Asymmetric key and digital certificates: SSC.

<table>
<thead>
<tr>
<th></th>
<th>GBA</th>
<th>SSC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td>Medium security</td>
<td>High security</td>
</tr>
<tr>
<td><strong>Disadvantages</strong></td>
<td>Less security that asymmetric key systems</td>
<td>Higher computational load A PKI infrastructure is needed to generate certificates</td>
</tr>
</tbody>
</table>