3GPP TSG SA WG3 Security — S3#22 **S3-020105**

25 - 28 February 2002

Bristol, UK

Source: **Nokia**

Title: **Public key certificates for cellular subscribers**

Document for: **Discussion**

Agenda Item: **7.6**

1. Overview

Emerging commercial digital services need a global business infrastructure for authorization, charging, settlement. Examples of such services include low- and medium- value payments, authorization for access to location information of mobile users, and authorization to push information to mobile users. The main challenge in building such a new infrastructure is the investment required and the difficulties in setting up new contractual agreements covering many countries, jurisdictions, and administrative domains. On the other hand, a cellular business infrastructure like in the existing GSM or future UMTS networks is nearly global in scale. Even though this infrastructure is currently limited to the authorization and settlement of just telecom services.

Many of these emerging services will be provided by parties that are not necessarily trusted by the operators or cellular subscribers. Therefore technical means to deal with, and preferably minimize, disputes between subscribers and service providers is necessary. Authorization of transactions using digital signatures is one technical means for reducing disputes.

We propose a procedure for issuing subscriber certificates to User Equipment (UE) and the parameters needed for that procedure. This procedure is run between the UE and the Core Network (CN) of the visited network. The CN contains the Certification Authority (CA) functionality. The certificate request, or more precisely the registration of the public key, is authenticated using the UMTS integrity key (IK). Because the registration is based on the existing USIM authentication, the issuing procedure is relatively easy and the subscriber certificates can be short-lived. The verifier can check if the subscriber certificate is valid by checking the certificate's validity period and the certificate status online with CA using, e.g. OCSP [OCSP].

Subscriber certificates issued in this manner are *authorisation certificates*. They do not certify an identity. Therefore the content of the Distinguished Name field in the certificate is not security-relevant. It can be any unique label, assigned by the operator CA.

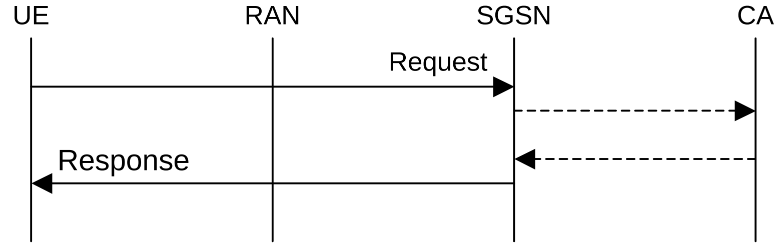

Figure 1. Certificate retrieval.

The expected functionality is the following:

UE should be able to send its public key to the visited network and receive either a subscriber certificate or a URL to the certificate in reply. UE may send informational attributes as well as proposed values of some certificate attributes to the CA (but the CA may freely override them). These are:

- key-origin: the level of trust in the public key, implied by the source from which it was obtained (e.g., from UICC, from another security module, from outside etc.)
- intended-key-usage: intended manner in which the signing key will be used on the UE. This is a boolean attribute indicating whether the user sees or not what is signed. This information can be used by the CA to set the values of any relevant attributes in the certificate. By default, the user should see what is signed. This parameter is typically set by application in the phone.

Optionally, UE may send a device certificate (i.e., a certificate issued by the manufacturer of the device containing the private key) with the request. The rationale for this certificate is that it can help increase the level of trust that the operator has in the submitted public key. Consequently, operator may issue a certificate with a greater level of authorization than otherwise.

Optionally, UE may also ask for the operator certificate (self-signed, or issued by some CA). This should be done only when necessary (i.e., only if the UE does not already have the certificate). The rationale for this certificate is that it can help the UE verify the public keys of its peers: for example, an operator certificate may be used as the root certificate to do server authentication during a TLS session [TLS]. As another example two UEs, each belonging to a different subscriber could use operator's certificates to verify each other's subscriber certificate for authorization purposes.

Therefore we introduce two protocols between UE and CN: subscriber certification and operator certificate retrieval.

The signalling channel is a scarce resource. Consequently, the following requirements arise:

1. the size of the messages sent over the signalling channel must have an upper bound (e.g., a certificate chain of unbounded number of links is not suitable)

2. in the normal case, the size of these messages must be as small as possible

2. Protocols

In this document we use the typographic convention:

1. Names of information element fields in protocol messages are underlined.
2. Names of types are *italic*.
3. Optional fields are marked "(OPTIONAL)"
4. Message parts marked "(CRITICAL)" require integrity protection. Message parts marked "(NON-CRITICAL)" do not require integrity protection. See more discussion on this aspect below.

Definitions of composite types mentioned in this section can be found in the X.509 based RFC [CERT-FORMAT] unless other reference is explicitely specified.

Both protocols are of a simple request/response type as shown in Figure 1.

2.1 Subscriber Certification

Request: (CRITICAL)

- key-info*: choice* of
  - public key: *SubjectPublicKeyInfo* (algorithm identifier, and bitstring)
  - pk_hash: *KeyIdentifier* (octet string; algorithm is SHA-1*)*
- key-origin: *byte*, with e.g. the following reserved values
  - 0 = from UICC, 1 = from another security module on UE, 2 = from outside UE, 3 = from UE own memory
- intended-key-usage: *Boolean* flag describing which usages are proposed, with the following values: 0 = automatic signing allowed 1 = signing with explicit user confirmation only
- user-plane-continuation-capability: *Boolean* flag. Should be set to true only if the terminal can accept a continuation URL (see the Response definition below).
- device-certificate: *Certificate* (OPTIONAL); certificate issued by the manufacturer of the device where the private key resides (e.g., a smartcard).

Response: (NON-CRITICAL)

- cert-info: *choice* of
  - subscriber certificate: *Certificate, WAPCertificate* [WAPCert]

- <u>subscriber certificate URL</u>: *URL* formatted as specified in [WPKI, section 7.3] from which the certificate can be retrieved. The UE will give this URL to the verifier instead of its certificate

- <u>failure</u>: *sequence* of

    - <u>error</u>: *byte*, with e.g. the following reserved values

        1. unknown cause

        2. continuation requested (continuation URL must be present below)

        3. service not available (this network does not issue certificates)

        4. service not available now (try later)

        5. service not possible without user-plane continuation (if terminal indicated user-plane-continuation-capability=false)

        6. key-origin not acceptable

        7. device certificate required (resend certificate request with the device certificate attached).

        8. device certificate invalid (e.g., expired, incorrect, or otherwise invalid)

        9. …

    - <u>continuation URL</u>: *URL* (OPTIONAL)

2.2 Operator certificate retrieval

When a successful response for the subscriber certification request is received, UE will find the Issuer Name of the operator CA. It can use this to check if it already has a valid certificate for the operator CA's public key. If not, it can initiate the operator certificate retrieval protocol below.

A second scenario for operator certificate retrieval is when a service provider specifies the operator (by e.g. specifying the hash of the operator CA's public key) in application-layer signalling. In this case, the UE will know the key hash of the operator but may not know the Issuer Name.

The operator certificate retrieval protocol is as follows:

Request: (NON-CRITICAL)

- <u>target</u>: (OPTIONAL) *choice* of

    - *Name*. Distinguished name of the issuing operator.

    - *KeyIdentifier* (octet string, algorithm is SHA-1*)* of the operator CA's public key

- user-plane-continuation-capability: *Boolean* flag indicating if the terminal can accept a URL of the operator certificate or not.

Response: (CRITICAL)

- operator-cert: *X.509v3 certificate*
- failure: *sequence* of
    - error: *byte*, with the following reserved values
        1. unknown cause
        2. no matching certificate
        3. service not available now (try later)
        4. service not possible without user-plane continuation (if terminal indicated user-plane-continuation-capability=false)
        5. …
- operator cert-info: (OPTIONAL) *sequence* of
    - hash: *KeyIdentifier* (octet string, algorithm is SHA-1)
    - url: *URL* of the operator certificate

3. Notes

3.1 Subscriber certification protocol

- Why not use PKCS-10 or IETF CRMF?

  There are no mandatory requirements for client registration in traditional PKI technical infrastructure because it could happen over a variety of transports, including non-electronic means. Thus, there is no de-facto standard to follow. There are two possibilities: PKCS10 by RSA [PKCS10v1.7] and RFC 2511 [CRMF] by the IETF pkix working group. PKCS10 appears to be more widely implemented and used. They are similar. One notable difference is that proof-of-possession is optional in RFC 2511.

  We have two reasons for not using a standard certificate request message format. By using the format as described above in Section 2 we

  1. can keep the message sizes small and bounded.

  2. have the flexibility of using the information elements needed in the cellular subscriber certification case (Section 2.1), without having to standardize new extensions to PKCS-10 or CRMF.

- How is the user-plane continuation-capability flag used?

  The terminal indicates with the user-plane-continuation-capability flag if it knows how to handle continuation URLs. If user-plane-continuation is set to false, the CA MUST NOT return a continuation URL. If the request contains a PK-hash rather than the full public key, user-plane-continuation MUST be set to true.

- How to specify restrictions on the use of the certified public key?

  The CA may map the intended-key-usage flag to appropriate certificate attributes (e.g., keyUsage or extKeyUsage) or certificate types (e.g., WTLS certificate vs. a full X.509v3 vertificate).

  X.509v3 certificate supports two classes of certificate extensions: *keyUsage* and *extKeyUsage*. Some of these, like cRLSign, are not applicable to subscriber certificates.
  The *extKeyUsage* extension allows the possibility for any organization to define additional key usages. However new *extKeyUsage* extensions would require a new object identifier for the new type. Potential new type s are:

  - onLineCheckRequired: *Boolean*, to indicate to the verifier that it should perform an on-line certificate status check before accepting signatures with respect to this certificate.

  - maxAuthorisationAmount: *amount* and *currency*, to indicate to the verifier that this certificate can be used for off-line authorisation of individual payment transactions up to the specified amount. (i.e, if the transaction amount is above the specified limit, the verifier SHOULD either perform an on-line authorisation check or disallow the transaction.

As a general rule-of-thumb, we would like to minimize the need to define new ExtKeyUsage extensions in this standard. Operators can of course define and use new ExtKeyUsage extensions without standardizing. The terminal won’t understand such extensions. Service Providers need to understand them.

There are no certificate extensions defined for WTLSCertificate.

See [CERT-FORMAT] for more information on *keyUsage* definitions.

- What about proof-of-possession?

  Without proof-of-possession we have to restrict the usage of subscriber certificates. For example, “for all applications that intend to use subscriber certificates it should be checked that proof-of-possession is not required”. This requires further study.

  There are two ways to support proof-of-possession: it can be done via user-plane continuation, as described in Section 2.1 or proof-of-possession can be added to the Request message. The former case is not an option if the general approach of user-plane continuation is not feasible. The latter case implies that the size of the subscriber certification Request message over the signalling-plane will be rather large (at least 300 bytes for a 1024-bit RSA keypair).

3.2 Operator certificate retrieval protocol

- How the user-plane continuation will be used?

  The critical object in operator certificate retrieval is the operator certificate itself. If user-plane continuation is used, the Reply in the signalling-plane contains only a hash of the operator certificate along with a URL where the full certificate can be retrieved. UE can retrieve the full certificate via the user-plane, and MUST check it against the hash received via the signalling-plane.

4. Open issues

4.1. Subscriber certification protocol

- What additional attributes are needed in the Request?

    - proposed-duration field in request: UE can propose duration for the certificate. Type can be two bytes, specifying minutes.

- Should UE be able to propose a Common Name part of the Distinguished Name for the certificate?

    As described, the DN in the certificate has no security relevance. But it may be a convenience.

- Is the user-plane continuation approach technically feasible?

    The problem with user continuation is that it is not trivial in UE to relate a signalling-plane event to user-plane event. If there are no restrictions on the bandwidth in the signalling-plane, user-plane continuation is not necessary. The reason for introducing user-plane continuation is that signaling plane has limited bandwidth and certificate request/response messages are relatively large. The user-plane has enough bandwidth, but user-plane messages are not automatically protected by IK.

    However, only critical objects need to be protected by IK. The idea behind user-plane continuation of certrificate retrieval is that a long critical object can be replaced by a short cryptographic hash when sent over the signaling plane; the object itself, and other related non-critical objects (i.e., objects that do not require protection by IK) are then sent during the user-plane continuatoin. The messages sent over the user-plane are securely linked to the messages sent over the signaling plane.

    The continuation URL is used to trigger the transfer of non-critical data over the user-plane, between CA and UE. When UE receives a continuation URL, it should go to that URL (presumably by triggering a browser). What happens after that depends entirely on what is sent back. Example possibilities are:

    - Proof-of-possession interaction: server sends back a WML deck with a nonce, as described in the sample certificate request in Section 7.3.4 of [WPKI].

    - Subscriber certificate with approprite MIME type (e.g., application/x-x509-user-cert) so that UE can store the certificate in the appropriate place: the operator CA may choose to do this if it was not possible to send the entire certifiate over the signalling-plane (e.g., because of congestion).

- Should the intended-key-usage attribute in the Request be more general?

    The following options are possible:

    - Allow the possibility of including a full *keyUsage* vector (i.e., the way it would appear in a certificate), and optional *extKeyUsage* objects in the request.

- Allow a way for UE to say what type of certificate is requested (WTLS, X.509v3, or WAP).

4.2 Operator certificate retrieval protocol

- Should it be possible to get operator cert signed by some other CA also ?

More specifically, should the following to be added to the operator cert retrieval request:

- Issuer of the CA certificate: *Name*. Distiguished name of the CA that issued the CA certificate for the operator (OPTIONAL)

5. Message payload sizes

For 1024-bit RSA keys, and SHA-1 as the hash algorithm, we estimate typical sizes for cryptographic objects as follows:

public key = 150 bytes

signature = 150 bytes

certificate = 900 bytes

hash = 30 bytes

Size of a Distinguished Name or Issuer Name string = 50 bytes

Approximate sizes of the message payloads in bytes are as follows. Figures in italics indicate the case if user-plane continuation is possible.

	Request		Reply	
	Maximum	Minimum	Maximum	Minimum
Subscriber Certification	150 + 900	150 *30*	900	50 (WPKI URL)
Operator Certificate Retrieval	50 + 50	0	900	900 *30 + 50*

References

[CERT-FORMAT] Internet X.509 Public Key Infrastructure Certificate and CRL Profile, RFC 2459.

[CRMF] Internet X.509 Certificate Request Message Format, RFC 2511.

[OCSP] Internet X.509 Public Key Infrastructure Online Certificate Status Protocol - OCSP, RFC 2560.

[PKCS10v1.7] PKCS #10, v1.7: Certification Request Syntax Standard, RSA Laboratories, May 26, 2000. Also issued as RFC 2986.

[TLS] The TLS Protocol Version 1.0, RFC 2246.

[WAPCert] WAP Certificate and CRL Profiles, WAP-211-WAPCert, Version 22-May-2001.

[WPKI] Wireless Application Protocol: Public Key Infrastructure Definition, WAP-217-WPKI, Version 24-April-2001.

[WTLS] Wireless Transport Layer Security, WAP-261-WTLS-20010406-a, Version 06-Apr-2001.