**3GPP TSG SA WG3 Security** **S3-020541**

**8th – 11th October, 2002**

**Munich, Germany**

| | |
|---|---|
| **Agenda Item:** | 7.7 |
| **Source:** | E-mail discussion chairman |
| **Title:** | Conclusions on Proof of Possession discussion |
| **Document for:** | Discussion/Decision |

The discussion was kicked off in August but got more drive in the beginning of September with the attached input paper. The main comments of the discussion are marked using “track change”-tool into the paper.

As a general conclusion there seemed to be consensus that PoP would be useful only in protecting against “sloppy application designers”. On the other hand, no consensus was reached on whether we should aim to achieve also this kind of protection by standardization.

**3GPP TSG SA WG3 Security**

**Email discussion input**

---

| | |
|---|---|
| **Source:** | **Nokia** |
| **Title:** | **Proof of Possession: an overview** |
| **Document for:** | **Discussion** |

---

# Proof of Possession: an overview

## Introduction

The purpose of this note is to study the rationale and requirements for proof of possession and the implications of requiring it as part of certificate requests. In particular, this note is intended to be used in SA3's current tasks related to the subscriber certificates work item (see item 1 in [S3-020447]).

## Use of a private key

There are two primary ways in which a private key (of an asymmetric key pair) is used:

- **commitment**: By signing a message with the private key, the purported controller of the private key commits himself to the signed message. Enabling ***non-repudiation*** is an example application of this kind.

- **claim**: By signing a message with the private key, or by decrypting a ciphertext (and demoinstrating knowledge of the plaintext), the purported controller of the private key can stake a claim for a benefit or ownership . For example, if the signature is on a plaintext (such as a message m) and the signature is accompanied by an identity certificate for me (an entity X), the claimed fact may be "X is certified to have control of this private key, and therefore message m signed by this key belongs to my identity (i.e., m is known to me and uttered by me)". In other words, message **authentication** is an example application of this kind.

- **encryption**: Of course, a keypair can also be used just for encryption. This seems to have similar properties like the "claim" use case.

In standard X.509v3 certificates, it is possible to use the `keyUsage` or `extKeyUsage` parameters to indicate the types of uses (e.g., "non-repudiation") to which the public key certificate is limited.

## What is Proof of possession (of private key)?

At the time of enrolment of an end entity in a PKI, it may be required to prove that it knows the private key corresponding to a claimed public key and that it controls the use of this private key. This is commonly referred to as the "proof of possession (PoP)".

The CMP RFC (RFC 2510, currently being revised [RFC 2510bis]) states that PoP is necessary because "in order to prevent certain attack and to check the validity of the binding between and end entity and a keypair" (Section 2.3 of [RFC 2510bis]). It does not describe what these attacks may be or whether they are limited to the case of "identity certificates" only.

The WPKI specification [WPKI] states that PoP is necessary "In order to avoid certain substitution attacks" (Section 4.1) but does not describe the attacks themselves.

The PKCS #10 specification [PKCS10] states (in Section 3, Note 2) "The signature on the certification request prevents an entity from requesting a certificate with another party's public key. Such an attack would give the entity

the minor ability to pretend to be the originator of any message signed by the other party. This attack is significant only if the entity does not know the message being signed and the signed part of the message does not identify the signer. The entity would still not be able to decrypt messages intended for the other party, of course."

The rationales and implications of PoP have been discussed in standards meetings and mailing lists. Yet, there does not appear to be any easily available or commonly known papers or articles that document these issues. It appears to be yet another case of undocumented folklore within the communities involved.

Stuart Ward: The ultimate requirement is that when mobile signature is used that there is a audit path from that signature back to the person. By this I mean the identifying details used at registration and the subsequent transaction history. This is what retailers and banks want to they know they are going to get paid.

Asokan: You are right about the ultimate requirement. Signatures are backed up by subscriber certificates. Subscriber certificates are issued subject to AKA authentication. So it will be possible to trace a signature (used with a subscriber certificate) back to a cellular subscription. The critical enabler here is AKA authentication. PoP is not needed for this, and won't make any difference in this respect.

Whether a subscription can be traced to a person depends on how the operator creates the subscription. Here, too, PoP does not make any difference.

Stuart Ward: Does PoP provide a vital step in that audit trail?

It seems to me that it could help provide this, but to see if it is really necessary we need to examine the complete registration process rather than look at just a single step. The one thing that is different in the Mobile signature is that we have an existing security association between the network (AuC) and the USIM (or SIM) so the registration process might a lot simpler that the standard Internet scheme.

Asokan: You are exactly right: subscriber certificates is a means of leveraging the existing cellular authorization infrastructure to extend authorization sevices to third party service providers. This is the central idea. This is what can give cellular opreators an advantage over someone else who attempts to build an authorization infrastructure from scratch.

## Security of enrollment

The enrolment process (i.e., submitting a public key to a CA/RA and requesting a certificate) must be authenticated. Typically this is done by distributing a shared one-time key or password a priori (e.g., by mailing scratch cards containing initial PIN codes to potential users of PKI). But it can also be authenticated by other means available: such as any existing or derivable security association between the end entity and the certifier, e.g., as done in [PIC]. The latter approach is applicable when we need to bootstrap a PKI from an existing infrastructure as in the case of 3GPP subscriber certificates. The purpose of this authentication is to allow the CA/RA to determine whether the certificate request is to be approved.

## Attacks not prevented by PoP

A certificate request contains a claimed public key, and any additional information ( e.g., a claimed identity, additional certificates in support of the request). In some cases (e.g., obtaining an attribute certificate), the request may contain a Distinguished Name instead of a public key. We do not consider this case because use of such certificates should be accompanied with an identity certificate that binds a public key to the said name. Therefore PoP does not appear to be necessary when requesting attribute certificates.

The certificate request is secured as mentioned in Section 0. PoP is *not* intended as a replacement for having to design appropriate mechanisms for the security of enrolment.

If PoP is required as part of certificate request, it assures the CA/RA that the requestor had access to the private key corresponding to the public key on which the certificate is requested.

Suppose the private key is used for commitments only (such as for non-repudiation) If PoP is not done during the certificate request process, a legitimate end entity **A** can indeed obtain a certificate binding *his* name to the public key of another end entity **B**. But this only leaves **A** (the attacker) liable for commitments made by **B** (who controls the private key and can make signatures).

Note that **A** still cannot send an arbitrary message m and claim that it belongs to and/or was sent by **B**. The security of the enrolment process (Section 0) is intended to prevent **A** from being able to send arbitrary signed message and fool the attacker into thinking that they came from **B**. See Section 1.1 for more discussion.

## Attacks prevented by PoP

### 1.1 Replacing public key in enrolment request sent by victim

#### 1.1.1 Description

In a certificate request sent by **B**, the attacker replaces **B**'s public key by **A**'s public key.

#### 1.1.2 Assumptions

1) *Security of enrolment is weak*: Normally, this attack is prevented by the mechanism for securing the enrolment. But this may not be strong enough in various ways:

   a) the cryptographic transforms used are inappropriate and allows an attacker on the network to change the contents of the request or fake a new request without being detected. We can call this the *"weak password attack"*.

   b) the access control mechanisms on the victim's device are not robust enough that the attacker is able to (i) change the contents of the local public key repository without being detected, or (ii) access the secret keys used to ensure the security of enrolment. We can call this the *"virus attack"*.

2) *Although the attacker is able to insert a public key into the victim's device, he is unable to insert a private key or intercept the communication path between the signing algorithm and the calling application*: if this were not true, **A** would insert a whole key pair into **B**'s device. Thereafter PoP is of no use. Note that **A** need not suffer any harm by inserting a private key: it can be the private key of a key pair that **A** generated solely for this attack. Alternately, even if **A** cannot insert the private key in the standard place where private keys are stored on the device (e.g., WIM card), it is enough if **A** intercepts private key operation requests or replaces the replies.

1.1.3 Type of uses

In *commitment* type uses

- if the public key is a signature verification key, then **A** can make commitments (signatures), have them supported by the certificate, and leave **B** liable for them.

In *claim* type uses,

- if the public key is an encryption key, then **A** can send it to a peer and trigger the peer into encrypting, with this key, some confidential data intended for **B.**

1.1.4 Extent of protection

In either case any protection provided by PoP is subject to *both* the assumptions listed in Section 1.1.2 being true.

## 1.2 Using victim's public key in own enrolment request

1.2.1 Description

In a certificate request sent by **A**, it can use the public key of another legitimate end entity **B**.

1.2.2 Assumptions

The basic attack described in the next section does not appear to rely on any strong assumption other than badly designed applications or application protocols.

1.2.3 Type of uses

Without PoP, **A** will be issued a certificate containing **B**'s pubic key.

In *claim* type uses

1) if the public key is a signature verification key, then **A** could falsely claim ownership of messages that were signed by **B**. This is the attack described in PKCS 10 specification (see Section 0). Note that if the application protocol would bind some identification of the sender within the signed message, and the verifying party's application checks this identification, then this attack would not work. We can call this the *"sloppy application protocol attack"*.

2) if the public key is a signature verification key, then **A** can mislead a third entity **C** into revealing to him some private data intended for **B.** This works as follows [WMI].

**A** obtains a certificate for **B**'s public signature verification key. Then **A** waits until **B** sends a signed, encrypted message and **B**'s certificate to **C**. **A** intercepts these. A does not know what the contents of the message were. **A** then forwards this intercepted message to **C** along with the certificate he obtained earlier. **C** concludes that the message actually came from **A** because all cryptographic checks succeed. This might cause **C** to reveal some private information to **A** (e.g., something about the contents of the message he just received).

In *commitment* type uses, there does not appear to be a way in which this can be used. However, there may be a software program like a Mail User Agent (MUA) that does use public keys in both ways (commitment and claim). If the application is not properly written, it may either not pay attention to the limited use of a certificate (such as `keyUsage` set to "non-repudiation"), or if the user is careless, then he might make be led to an incorrect conclusion and act based on it. For example, in example 2), suppose the certificate has `keyUsage` set to "non-repudiation", but the MUA of **C** does not indicate this clearly. So **C** may be misled into assuming that the signature and certificate authenticate the sender, and make the same conclusions as though the certificate is also intended for "claim" type of uses. We can call this the *"sloppy application attack"*.

### 1.2.4 Extent of protection

The attack against "claim" type application assumes that application software and application protocol designers may have made some basic mistakes. The security of the users of such applications can benefit from mandatory PoP. The attack against "commitment" type application can also be avoided by if

a) a key pair has only one type of use (i.e., not authentication *and* non-repudiation simultaneously), and

b) the software application of the relying party handles `keyUsage/extKeyUsage` restrictions correctly.

If it is difficult to ensure either of the above (e.g., it may be difficult to mandate the former, and unrealistic to expect the latter) then PoP can help limit the damage. However, attempting to provide protection against sloppy application designers is ultimately a doomed exercise.

Stefan Schröder: to my understanding the paper's essence says that PoP is only useful to counter weak security of the enrolment path (1.1.2).
But in the proposed architecture alternatives, we have strong user authentication and a protected enrollment path (UMTS AKA took place before). (Ok, protection of the path RNC-CA is currently not specified and up to the operator.)

The other problems "sloppy application" and "virus" can not fully be countered due to their nature.

The quite practical misuse case 1.2.3 2) may also be countered by a check in the CA if the submitted public key is already registered for another entity. PoP may be the more elegant solution, but it also requires more roundtrips.

So I agree that PoP should not be needed.

A major discussion point the paper mentions but also leaves open is liability: who will be liable for disputed transactions backed by subscriber certificates? HN operator, SN (=CA) operator, service provider, application developer, subscriber? The usage scenario "payment" suggests that the HN is liable for a certificate issued by the SN. I think this scenario calls for MAPSec to avoid fraud using eavesdropped authentication vectors.

Asokan: The question of who assumes liability is closely related to the trust assumptions in the system. Retaining the same trust assumptions and procedures that are already designed/used would help to quickly deploy services based on subscriber certificates.

So, it would be a good idea to handle disputes related to subscriber certificates exactly in the same way as how disputes about cellular servics are handled today.

Question: who is liable today if a subscriber disputes a charge about an alleged call he made while roaming: the subscriber, HN operator, or SN operator?

Stefan S:
This is something we probably cannot settle in standardization but which may prevent a good idea from being used.

Asokan: You are probably right that this cannot be settled in standardization.
But note that even if this turns out to be a problem, it is only relevant to monetary transactions. Subscriber certificates have lots of other applications which do not involve money.

Marc Blommaert: But may PoP be dropped now, relying on the argument that PoP is only good to
prevent harm to badly designed application? The decision on it is probably more complex.
In my point of view the adoptation of PoP may be necessary to avoid attackers being able to take advantage of badly designed applications. 3GPP may not be able to analyse all application designs that will be using subscriber certificates (is it even the task of 3GPP or OMA ?). However, as subscriber certificates will be issued taking advantage of the 3GPP subscriber infrastructure, any bad publicity caused by flaws and subsequent fraud will be bad for the 3GPP-image, and may hinder the growth of M-commerce.

Marc Blommaert: It is basically said that application programs will be flawed anyhow, and there is no way to stop this. I cannot see how the CA would in this way assume any liability for flaws in application programs which cannot be stopped.

### Does PoP do any harm?

PKI standards (such as PKIX and WPKI) seem to mandate PoP. Technically requiring PoP is safe since it does not seem to do any harm and it is clearly needed for at least one class of applications (the “ownership claim” class). But there may be other concerns.

*Bandwidth*: PoP means that the requestor has to perform a private key operation. This in turn implies that a large message (e,g., signature or encryption) needs to be sent between the requestor and the certifier.

*Latency*: To prevent against replay attacks the PoP protocol must ensure freshness. As usual, this can be done using timestamps (which requires synchronized clocks) or better still, the certifier can send a challenge nonce. But this means that the certificate request procedure will contain at least 3 messages (instead of being just request/response).

*Novel applications*: It appears that the need for PoP arose from the "traditional" PKI scenarios where the certificates issued are identity certificates. In such cases, it is of course quite logical to try to prevent two persons from attempting to get certificates for the same public key. However, as limited scope PKIs are become more realistic and more prevalent, there may be new uses that are prevented or made harder by mandating PoP. For example, suppose certificates are used to enable operator billing. A use case may be for **A** to obtain a certificate for **B**'s public key that allows **B** as a surprise gift voucher (so that **B** is allowed to spend a certain amount of money which will be billed to **A**). If PoP is required, then obtaining such a certificate will require **B**'s involvement, which will eliminate this use case!

In summary, if the environment is such that the communication channel between the requestor and the certifier has no bandwidth or latency constraints, it is safe to require PoP. Otherwise, it may be necessary to do PoP only where necessary.

## Other rationales for justifying PoP

One of the primary intents being PoP seems to be to minimize potential damage due to

a) badly designed application-level protocols, or

b) badly designed end entity application software, or

c) carelessness of an end entity user

The first assumption in Section 1.1.2 or the second attack in Section 1.2.3 may be due to one of the factors listed above. PoP would eliminate or at least reduce the likelihood of the resulting attacks. CA operators therefore may view PoP as a way of protecting them from liability arising from damage resulting in these factors.

However, the following should be noted: Currently, there is no way of looking at a certificate and knowing if PoP was required before issuing it. So any good application developer has to assume that PoP was not done.

## Should there be PoP in 3GPP subscriber certification?

Whether PoP is to be included in the 3GPP subscriber certification procedure depends also on two other considerations:

- The chosen architecture: [S3-02037] describes four architecture alternatives. The characteristics of the secured communication channel between UE and the CA depends on which architecture is chosen. In

general, communication channels involving a mobile device are bandwidth limited. But If the SGSN-based solution is chosen, then the communication channel is the signalling plane which is severely bandwidth-constrained.

- The chosen protocol: If an existing certificate request protocol (such as PKCS-10) is chosen, then it may already have support for PoP. Of course, this choice itself is affected by the previous consideration.

In the original SGSN-based proposal [S3-020105], PoP was mentioned as an optional feature (it was **not** precluded). The last bullet item in the list in Section 3.1 of [S3-020105] describes how PoP can be done even in this architecture.

## Conclusion

Many standards allude to unspecified "attacks" in justifying why PoP is needed. PoP may be necessary only for certain classes of applications. In particular, enabling non-repudiation does not seem to require PoP.

Based on the above analysis, we see three options for 3GPP:

1. mandate PoP in all cases; This has the side effect that at least some use cases will be prevented.

2. mandate PoP at least when use of key includes *anything other than* the commitment type. This assumes that application developers will correctly check the `keyUsage` parameter *and* clearly indicate it to the user.

3. do not require PoP at all. This assumes that the certificates will be used with applications and application protocols which are well designed (3GPP specifications can clearly indicate what "well designed" means in this context).

Our recommendation is the last option for the following reasons:

- trying to provide protection from badly designed applications or application protocols is not advisable: it might lead to the CA operator being liable for *any* design error made by arbitrary application developer.

- the specification should not have a feature that does not have a compelling reason; the only reasons we found so far in the above discussion is basically "protecting potential victims from badly designed applications or application protocols".

## References

[RFC 2510bis] C. Adams and S. Farrel, "Internet X.509 public key infrastructure: Certificate management protocols", IETF PKIX working group Internet draft `draft-ietf-pkix-rfc2510bis-06.txt`, December 2001.

[WPKI] WAP forum "Wireless application protocol, public key infrastructure definition", WAP-217-WPKI, 24 April 2001.

[PIC] Y. Sheffer et al, “PIC, A Pre-IKE Credential Provisioning Protocol”, IETF ipsra working group Internet draft `draft-ietf-ipsra-pic-05.txt`, Februrary 2002.

[S3-02037] Nokia, “Architecture alternatives for supporting subscriber certificates”, July 2002

[S3-020105] Nokia, “Public key certificates for cellular subscribers”, February 2002

[S3-020447] SA3 Liasion Statement to SA1 and SA2, “LS on architecture and requirements for subscriber certificates”, July 2002.

[WMI] Michael Waidner, Personal communication, August 2002

[PKCS10] RSA laboratories, PKCS #10 v1.7: Certification Request Syntax Standard, May 26, 2000 (a previous version [1.5] is RFC 2314)