CA Generation of EE Private Keys

Status: This document is for discussion only and does not represent TAGPMA or IGTF policy.

Background

The Classic Profile states, "The authority shall issue X.509 certificates to end-entities based on cryptographic data generated by the applicant, or based on cryptographic data that can be held only by the applicant on a secure hardware token." The MICS and SLCS profiles contain similar language. The profiles also state that private keys "must be generated and stored in accordance with the currently approved version of the Guidelines on Private Key Protection" which allows four methods for private key generation:

1. Inside a secure hardware token
2. Locally on an appropriate computer system of which the user is the sole user and administrator
3. On an appropriate computer system that is administered by the users home organisation
4. On an appropriate computer system that is administered by a third party

with specific requirements for each method. In particular, method 4 requires:

- Key material MUST only be generated in the system as a result of a user action.

While the term "third party" is not defined in the Guidelines on Private Key Protection, we try to capture the current understanding in the following definition:

- **third party**: an entity physically separate from the CA (i.e., operated using separate hardware to which CA operators do not have administrative access), located either in the same or a different organization from the CA (i.e., there is not a requirement for the CA and third party to be located in separate legal organizations)

It is unclear if the current language in the profiles ("generated by the applicant, or ... held only by the applicant") allows methods 3 or 4 for private key generation as documented in the Guidelines on Private Key Protection.

Motivation

The requirement that end entities generate their own private keys makes it more difficult for CAs to support certificate applicants who are PKI novices. Methods for key generation vary across web browsers (and different versions of the same web browser) and often result in a confusing experience for the applicant (for example, giving the user a choice of "Medium Grade" or "High Grade" keys or a choice of "Cryptographic provider"). An alternative is to have the applicant install OpenSSL or some custom software/script or launch a Java Web Start application (hoping they have Java installed), each of which present their own challenges and frustrations to the novice applicant.

This motivates us to examine the reasons we require end entities to generate their own private keys and consider whether we might allow CAs to generate private keys on behalf of applicants to simplify the application process.

Motivating Use Case

Our motivating use case for purposes of discussion is:

- The applicant using a web browser (for example: lynx) connects to the CA web site over HTTPS.
- The applicant authenticates to the CA web site (for example: with a password or an OTP card or SAML or Kerberos) over HTTPS.
- The CA prompts the applicant for a 12 character or longer private key passphrase over HTTPS.
- The CA generates a keypair and certificate for the applicant and delivers them (protected by the applicant-chosen passphrase) in a PKCS12 object to the applicant over the HTTPS session.
- The CA securely discards its copies of the PKCS12 object.
- The applicant loads the PKCS12 object into his or her browser or operating system certificate store or saves it to a file on disk as desired.

Key Repositories and Other Options

An alternate approach is to pre-load private keys into key repositories (such as MyProxy or Active Directory). For the purposes of this discussion, we consider this (and other) alternative approaches to be out of scope. Our goal is not to debate different approaches to key generation and delivery but instead to discuss whether the specific method of CA generated keys described in the motivating use case above is acceptable.

Risk Analysis

If the CA generates the private key instead of the applicant, risk to the private key is introduced at the CA and in transit from the CA to the applicant. Note that already we trust the CA to issue certificates appropriately, and we accept the risk that a misbehaving CA could issue a certificate to a new keypair to impersonate a subject (either through CA operator misbehavior or CA compromise). The new risk is that the specific private key could be misused by a misbehaving CA, so actions linked with that specific private key (rather than actions linked more generally to the certificate subject) are put at risk. However, in current practice in grids today, rights and privileges are associated with certificate subjects rather than specific keys, limiting the impact of this new risk.

> "rights and privileges are associated with certificate subjects rather than specific keys"
> This identifies one of the risks, a risk which has been built-in to grids. Other infrastructur
> may use the key as an attribute, or prevent a client from having more than one operational key
> pair of a given type at a time.
> (Michael Helm)

Regarding the risk of private keys in transit, the standards from the Guidelines on Private Key Protection apply. Specifically:

- The private key SHALL NOT be sent in clear text over a network.
- The pass phrase SHALL NOT be sent in clear text over a network.
- The encrypted private key file SHOULD NOT be sent over the network unprotected. Network protections may be via encryption, or by physical control of the network in a trusted environment.

These standards are met by the use of TLS between the subscriber's browser and the CA and the use of passphrase protection in the PKCS12 object.

The discussion in the Guidelines on Private Key Protection regarding the trade-offs for generating private keys on behalf of subscribers is also relevant:

> The aim of this document is to make a qualified trade-off between non-repudiation and current, actual practice. In the case of portals, or any third-party box, requesting certificates on the user's behalf based on key material generated or held by the third party, that box can use that key theoretically without the user's knowledge or consent. This document intends to provide guidelines for such systems, to countermand the loss of user binding by providing policies and operational guidelines.

> Another risk, which I don't think is mentioned here explicitly, is a fundamental attack on
> the assumptions inherent in digital signature - which is that the client is the only one
> able to use the private key linked to this certificate. Grids PKI have already bypassed
> the value of that assumption to some extent (see above) but they have not eliminated it.
> Our current profiles support this ... paradigm.
>
> Generating keys on behalf of subscribers in an enterprise is clearly appropriate - it's the
> enterprise's risk to take.
>
> Is it appropriate here? One of the risks that comes with this is no matter what is said here
> "The private key of key-pairs generated by ... are not held ... after being transferred to the
> that something else might happen instead. There are some hypothetical liability issues too
> (not hypothetical enough for me to be comfortable with them but unnecessary to go into here).
> It's not clear to me that all the inherent risks can be eliminated even if the 3rd party CA &
> 4th party storage/generator functions are separated.
>
> We have mixtures of federation and enterprise behavior - does this make a difference?
> (Michael Helm)

Proposal

If consensus is reached that CA generation of private keys is acceptable, we recommend the following changes:

- Change the "generated by the applicant, or ... held only by the applicant" language in the profiles to "generated and stored in accordance with the currently approved version of the Guidelines on Private Key Protection."
- Remove the qualifying phrase "that is administered by a third party" from the Guidelines on Private Key Protection section on generation of private keys so CAs are not excluded from key generation method 4.

to top

Edit | **Attach image or document** | **Printable version** | **Raw text** | **More topic actions**
Revisions: | r1.15 | **>** | **r1.14** | **>** | **r1.13** | **Total page history** | **Backlinks**

You are here: Main > CAgenEEkeys **to top**

Ideas, requests, problems regarding TWiki? Send feedback