Composite Crypto for Hybrid PKIs Deployment

https://datatracker.ietf.org/doc/draft-pala-composite-crypto/

Massimiliano Pala <m.pala@cablelabs.com> - CableLabs / OpenCA

Daniel Van Geest <Daniel.VanGeest@isara.com> - ISARA

The Problem

- **PKIs are used to provide scalability for Key Management.** One type of PKIs that is predominant for securing communications and data is based on the X.509 standard.
- With the definition of new algorithms (e.g. more efficient factoring techniques) and technologies (e.g., quantum-based computing devices) that might be available in the short future, **the need to provide an easy-to-deploy and efficient solution capable of providing multi-algorithms authentication is paramount**.

The Proposed Solution

- We defined a method to combine public/private keys into a single one by encoding it as a SEQUENCE of keys and a corresponding SEQUENCE of Signatures
- In particular we introduce two new building blocks that constitute the core of the proposed approach:
  - **compositePublicKey / compositePrivateKey,** and
  - **compositeSignature**
- The concept is then applied to many (all) PKIX data structures to allow the use of multiple algorithms in certificates, revocation information, key store, etc.

The Proposed Solution (cont.)

- This solution allows for what we define as a “deferred algorithm agility” – deploy infrastructures and data that are protected with multiple algorithms (e.g., RSA, EC, or one of QRC)
- All keys in the compositePublicKey structure are associated with the same identity (it is just a single compositePublicKey, after all), thus not requiring multiple certificates each of which with its own key
- Relying party might rely on a subset of algorithm to verify the PKIX objects and might use additional algorithms in the future when support for them is added to the software stack
  - This allows for hybrid PKIs to be deployed today

Example: Public Keys in X.509 Certificates

- Currently X.509 Certificates provides the possibility to include only one public key. In particular, the following specifications applies:

```
AlgorithmIdentifier  ::=  SEQUENCE  {
        algorithm               OBJECT IDENTIFIER,
        parameters              ANY DEFINED BY algorithm OPTIONAL  }

SubjectPublicKeyInfo  ::=  SEQUENCE  {
        algorithm               AlgorithmIdentifier,
        subjectPublicKey        BIT STRING  }
```

CompositePublicKey Details

- When the (“compositePublicKey”) value is used for the algorithm identifier, that means that the value encoded in the associated public key field (e.g., the (“SubjectPublicKeyInfo”) field in X.509 Certificates as defined in RFC 5280 [1]) contains a SEQUENCE of DER encoded public keys and associated parameters.
- The public key value is encoded as the DER representation of the SEQUENCE of the (“SubjectPublicKeyInfo”) for each public key that are present in the (“CompositePublicKey”).
- The DER encoding of the sequence of public keys is then placed in the subjectPublicKey field of the certificate with NULL parameters.

CompositeSignature Details

- Similar approach to the compositePublicKey/compositePrivateKey
- The (“signatureValue”) field (BIT STRING) is the DER representation of a (“CompositeSignature”) that is a SEQUENCE OF BIT STRING (“signatureValues”) where each (“signatureValue”) carries the information about one of the “basic” signatures
- When the (“compositeSignature”) value is used for the algorithm identifier, that means that the value encoded in the associated signature field contains multiple public keys and associated parameters
- The sequence of signatures contains the signatures generated with the compositePrivateKey, that means that EVERY “basic” private key corresponding to the public keys in the “signing” compositePrivateKey must be used to generate the compositeSignature

Validating CompositeSignature(s)

- When validating a compositeSignature, the relying party might not support all algorithms used in the corresponding compositeKey
- Therefore we require that the relying party:
  - MUST verify at least one basic signature in the compositeSignature SEQUENCE
  - SHOULD verify all signatures whose algorithm is supported
- If none of the basic signatures in the compositeSignature can be verified successfully, the relying party that is validating the signature should reject the compositeSignature entirely.

Applicability

- Our work addresses all aspects of the PKIs ecosystem: from certificates to revocation information, from private keys storage to cryptographic messages
- For some PKIX objects, the application of this new “algorithm” is straightforward as we have seen for certificates (e.g., CRLs, OCSP messages, etc).
- For some other PKIX object, the use of compositePublicKey, compositePrivateKey, and compositeSignature structures might require additional considerations (e.g., PKCS#8), but we do not foresee (at this time) any limitations that would require changes in specs

IPR Disclosure

- CableLabs wants to be sure that the idea will be widely available and free to implement for everybody
- In order to protect our work, we filed for IP on the core of the idea (sequence of keys and sequence of signatures) to ensure nobody will try to restrict its use
  - CableLabs has provided the IPR disclosure for the patent application that was filed early last year when we started to work on the idea.
  - The IPR statement is available at: https://datatracker.ietf.org/ipr/3481/ (royalty-free with reciprocity)
  - Please refer to the text for the specific provisions

Next Steps

- Provide the details around the encoding of the missing PKIX objects
- Complete the sections that still require text in order to get the document ready for review
- (If the WG is interested in pursuing the idea), when ready we can work towards the possible adoption of the document (or documents) and re-chartering to include the work item (maybe by the next meeting in Montreal... ?)