Workgroup: LAMPS
Internet-Draft:
draft-ounsworth-pq-composite-keys-01
Published: 12 February 2022
Intended Status: Standards Track
Expires: 16 August 2022
Authors: M. Ounsworth (Editor) M. Pala
Entrust CableLabs

Composite Public and Private Keys For Use In Internet PKI

Abstract

With the widespread adoption of post-quantum cryptography will come the need for an entity to possess multiple public keys on different cryptographic algorithms. Since the trustworthiness of individual post-quantum algorithms is at question, a multi-key cryptographic operation will need to be performed in such a way that breaking it requires breaking each of the component algorithms individually. This requires defining new structures for holding composite keys, for use with composite signature and encryption data.

This document defines the structures CompositePublicKey, CompositePrivateKey, which are sequences of the respective structure for each component algorithm. This document makes no assumptions about what the component algorithms are, provided that they have defined algorithm identifiers. The only requirement imposed by this document is that all algorithms be of the same key usage; i.e. all signature or all encryption. This document is intended to be coupled with corresponding documents that define the structure and semantics of composite signatures and encryption.

Status of This Memo

Table of Contents

1. Introduction

During the transition to post-quantum cryptography, there will be uncertainty as to the strength of cryptographic algorithms; we will

no longer fully trust traditional cryptography such as RSA, Diffie-Hellman, DSA and their elliptic curve variants, but we will also not fully trust their post-quantum replacements until they have had sufficient scrutiny. Unlike previous cryptographic algorithm migrations, the choice of when to migrate and which algorithms to migrate to, is not so clear. Even after the migration period, it may be advantageous for an entity's cryptographic identity to be composed of multiple public-key algorithms.

The deployment of composite public keys, and composite signatures and composite encryption using post-quantum algorithms will face two challenges

- Algorithm strength uncertainty: During the transition period, some post-quantum signature and encryption algorithms will not be fully trusted, while also the trust in legacy public key algorithms will start to erode. A relying party may learn some time after deployment that a public key algorithm has become untrustworthy, but in the interim, they may not know which algorithm an adversary has compromised.

- Backwards compatibility: During the transition period, post-quantum algorithms will not be supported by all clients.

This document provides a mechanism to address algorithm strength uncertainty by providing formats for encoding multiple public keys and private keys values into existing public key and private key fields.

This document is intended for general applicability anywhere that keys are used within PKIX or CMS structures.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are used in this document:

ALGORITHM: An information object class for identifying the type of cryptographic key being encapsulated.

BER: Basic Encoding Rules (BER) as defined in [X.690].

COMPONENT ALGORITHM: A single basic algorithm which is contained within a composite algorithm.

COMPOSITE ALGORITHM: An algorithm which is a sequence of two or more component algorithms, as defined in Section 2.

DER: Distinguished Encoding Rules as defined in [X.690].

PUBLIC / PRIVATE KEY: The public and private portion of an asymmetric cryptographic key, making no assumptions about which algorithm.

2. Composite Structures

In order for public keys and private keys to be composed of multiple algorithms, we define encodings consisting of a sequence of public key or private key primitives (aka "component algorithms") such that these structures can be used as a drop-in replacement for existing public key fields such as those found in PKCS#10 [RFC2986], CMP [RFC4210], X.509 [RFC5280], CMS [RFC5652], and the Trust Anchor Format [RFC5914].

This section defines the following structures:

* The id-alg-composite is an OID identifying a composite public key.

* The CompositePublicKey carries all the public keys associated with an identity within a single public key structure.

* The CompositePrivateKey carries all the private keys associated with an identity within a single private key structure.

EDNOTE 2: We have heard community feedback that the ASN.1 structures presented here are too flexible in that allow arbitrary combinations of an arbitrary number of signature algorithms. The feedback is that this is too much of a "footgun" for implementors and sysadmins. We are working on an alternative formulation using ASN.1 information object classes that allow for compiling explicit pairs of algorithmIDs. We would love community feedback on which approach is preferred. See slide 30 of this presentation: https://datatracker.ietf.org/meeting/interim-2021-lamps-01/materials/slides-interim-2021-lamps-01-sessa-position-presentation-by-mike-ounsworth-00.pdf

2.1. Algorithm Identifier

2.1.1. Composite Public Key

The Composite algorithm identifier is used for identifying a public key and a private key. Additional encoding information is provided below for each of these objects.

When using this algorithm identifier it is implied that all component keys MUST be used in an AND relation; any cryptographic operation using this composite public key MUST use the it as an atomic object and use all component keys. This mode has the strongest security properties and is RECOMMENDED.

There is an additional security consideration that some use cases such as signatures remain secure against downgrade attacks if and only if component keys are never used in isolation and therefore it is RECOMMENDED that component keys in a composite key are uniquely generated.

```
id-composite-key OBJECT IDENTIFIER ::= {
    joint-iso-itu-t(2) country(16) us(840) organization(1) entrust(11402
```

EDNOTE 3: this is a temporary OID for the purposes of prototyping. We are requesting IANA to assign a permanent OID, see Section 4.

2.1.2. Composite-OR Public Key

EDNOTE: This section was written with the intention of keeping the primary Composite OID reserved for the simple and strict mode; if you want to do either a simple OR, or a custom policy then we have given a different OID. We are still debating whether this is useful to specify at issuing time, or whether this is adding needless complexity to the draft.

The Composite-OR algorithm identifier is used for identifying a public key and a private key. Additional encoding information is provided below for each of these objects.

When using this algorithm identifier, component keys MAY be used in an OR relation meaning that any one of the component keys may be used by itself. Implementors may also define more complex processes and policies using this algorithm identifier, for expmple allowing some algorithms by themselves and others only in combination. This mode is provided for applications that need to issue long-lived composite keys in a way that allows for backwards compatibility or staged adoption of new algorithms.

```
id-composite-or-key OBJECT IDENTIFIER ::= {
    joint-iso-itu-t(2) country(16) us(840) organization(1) entrust(11402
```

2.2. Composite Keys

A composite key is a single key object that performs an atomic signature or verification operation, using its encapsulated sequence of component keys.

The ASN.1 algorithm object for composite public and private keys is:

```
pk-Composite PUBLIC-KEY ::= {
    IDENTIFIER id-alg-composite
    KEY CompositePublicKey
    PARAMS ARE absent
    PRIVATE-KEY CompositePrivateKey
}
```

EDNOTE 4: the authors are currently unsure whether the params should be absent (ie this structure simply says "I am a composite algorithm"), or used to duplicate some amount of information about what the component algoritms are. See Section 2.3 for a longer ENDOTE on this.

2.2.1. Key Usage

For protocols such as X.509 [RFC5280] that specify key usage along with the public key, any key usage may be used with Composite keys, with the requirement that the specified key usage MUST apply to all component keys. For example if a Composite key is marked with a KeyUsage of digitalSignature, then all component keys MUST be capable of producing digital signatures. id-alg-composite MUST NOT be used to implement mixed-usage keys, for example, where a digitalSignature and a keyEncipherment key are combined together into a single Composite key object.

2.3. Composite Public Key

Composite public key data is represented by the following structure:

```
CompositePublicKey ::= SEQUENCE SIZE (2..MAX) OF SubjectPublicKeyInfo
```

The corresponding AlgorithmIdentifier for a composite public key MUST use the id-alg-composite object identifier, defined in Section 2.1, and the parameters field MUST be absent.

A composite public key MUST contain at least one component public key.

A CompositePublicKey MUST NOT contain a component public key which itself describes a composite key; i.e. recursive CompositePublicKeys are not allowed

EDNOTE: unclear that banning recursive composite keys actually accomplishes anything other than a general reduction in complexity. In particular, with the addition of Composite (AND mode) and Composite-OR (OR mode), recursion actually allows full boolean expression. Is this valuable?

Each element of a CompositePublicKey is a SubjectPublicKeyInfo object for a component public key. When the CompositePublicKey must

be provided in octet string or bit string format, the data structure is encoded as specified in Section 2.5.

2.4. Composite Private Key

The composite private key data is represented by the following structure:

```
CompositePrivateKey ::= SEQUENCE SIZE (2..MAX) OF OneAsymmetricKey
```

Each element is a OneAsymmetricKey [RFC5958] object for a component private key.

The corresponding AlgorithmIdentifier for a composite private key MUST use the id-alg-composite object identifier, and the parameters field MUST be absent.

A CompositePrivateKey MUST contain at least one component private key, and they MUST be in the same order as in the corresponding CompositePublicKey.

2.5. Encoding Rules

Many protocol specifications will require that the composite public key and composite private key data structures be represented by an octet string or bit string.

When an octet string is required, the DER encoding of the composite data structure SHALL be used directly.

When a bit string is required, the octets of the DER encoded composite data structure SHALL be used as the bits of the bit string, with the most significant bit of the first octet becoming the first bit, and so on, ending with the least significant bit of the last octet becoming the last bit of the bit string.

In the interests of simplicity and avoiding compatibility issues, implementations that parse these structures MAY accept both BER and DER.

3. In Practice

This section addresses practical issues of how this draft affects other protocols and standards.

~~~ BEGIN EDNOTE 10~~~

EDNOTE 10: Possible topics to address:

*The size of these certs and cert chains.
~~~

*In particular, implications for (large) composite keys / signatures / certs on the handshake stages of TLS and IKEv2.

*If a cert in the chain is a composite cert then does the whole chain need to be of composite Certs?

*We could also explain that the root CA cert does not have to be of the same algorithms. The root cert SHOULD NOT be transferred in the authentication exchange to save transport overhead and thus it can be different than the intermediate and leaf certs.

*We could talk about overhead (size and processing).

*We could also discuss backwards compatibility.

*We could include a subsection about implementation considerations.

~~~ END EDNOTE 10~~~

3.1. Textual encoding of Composite Private Keys

CompositePrivateKeys can be encoded to the Privacy-Enhanced Mail (PEM) [RFC1421] format by placing a CompositePrivateKey into the privateKey field of a PrivateKeyInfo or OneAsymmetricKey object, and then applying the PEM encoding rules as defined in [RFC7468] section 10 and 11 for plaintext and encrypted private keys, respectively.

3.2. Asymmetric Key Packages (CMS)

The Cryptographic Message Syntax (CMS), as defined in [RFC5652], can be used to digitally sign, digest, authenticate, or encrypt the asymmetric key format content type.

When encoding composite private keys, the privateKeyAlgorithm in the OneAsymmetricKey SHALL be set to id-alg-composite.

The parameters of the privateKeyAlgorithm SHALL be a sequence of AlgorithmIdentifier objects, each of which are encoded according to the rules defined for each of the different keys in the composite private key.

The value of the privateKey field in the OneAsymmetricKey SHALL be set to the DER encoding of the SEQUENCE of private key values that make up the composite key. The number and order of elements in the sequence SHALL be the same as identified in the sequence of parameters in the privateKeyAlgorithm.

The value of the publicKey (if present) SHALL be set to the DER encoding of the corresponding CompositePublicKey. If this field is
~~~

present, the number and order of component keys MUST be the same as identified in the sequence of parameters in the privateKeyAlgorithm.

The value of the attributes is encoded as usual.

4. IANA Considerations

The ASN.1 module OID is TBD. The id-composite-key and id-composite-or-key OIDs are to be assigned by IANA. The authors suggest that IANA assign an OID on the id-pkix arc:

```
id-composite-key OBJECT IDENTIFIER ::= {
    iso(1) identified-organization(3) dod(6) internet(1) security(5)
    mechanisms(5) pkix(7) algorithms(6) composite(??) }
```

5. Security Considerations

5.1. Reuse of keys in a Composite public key

There is an additional security consideration that some use cases such as signatures remain secure against downgrade attacks if and only if component keys are never used in isolation and therefore it is RECOMMENDED that component keys in a composite key are uniquely generated. Note that protocols allowing public keys using the Composite-OR algorithm identifier will have a more difficult time preventing downgrade and stripping attacks and therefore it is RECOMMENDED to use the default AND mode unless the application has a strong need for backwards compatability and migration.

5.2. Policy for Deprecated and Acceptable Algorithms

Traditionally, a public key, certificate, or signature contains a single cryptographic algorithm. If and when an algorithm becomes deprecated (for example, RSA-512, or SHA1), it is obvious that structures using that algorithm are implicitly revoked.

In the composite model this is less obvious since a single public key, certificate, or signature may contain a mixture of deprecated and non-deprecated algorithms. Moreover, implementers may decide that certain cryptographic algorithms have complementary security properties and are acceptable in combination even though neither algorithm is acceptable by itself.

Specifying a modified verification process to handle these situations is beyond the scope of this draft, but could be desirable as the subject of an application profile document, or to be up to the discretion of implementers.

2. Check policy to see whether A1, A2, ..., An constitutes a valid combination of algorithms.

```
if not checkPolicy(A1, A2, ..., An), then
  output "Invalid signature"
```

While intentionally not specified in this document, implementors should put careful thought into implementing a meaningful policy mechanism within the context of their signature verification engines, for example only algorithms that provide similar security levels should be combined together.

EDNOTE 11: Max is working on a CRL mechanism to accomplish this.

5.3. Protection of Private Keys

Structures described in this document do not protect private keys in any way unless combined with a security protocol or encryption properties of the objects (if any) where the CompositePrivateKey is used (see next Section).

Protection of the private keys is vital to public key cryptography. The consequences of disclosure depend on the purpose of the private key. If a private key is used for signature, then the disclosure allows unauthorized signing. If a private key is used for key management, then disclosure allows unauthorized parties to access the managed keying material. The encryption algorithm used in the encryption process must be at least as 'strong' as the key it is protecting.

5.4. Checking for Compromised Key Reuse

Certificate Authority (CA) implementations need to be careful when checking for compromised key reuse, for example as required by WebTrust regulations; when checking for compromised keys, you MUST unpack the CompositePublicKey structure and compare individual component keys. In other words, for the purposes of key reuse checks, the composite public key structures need to be un-packed so that primitive keys are being compared. For example if the composite key {RSA1, PQ1} is revoked for key compromise, then the keys RSA1 and PQ1 need to be indivitually considered revoked. If the composite key {RSA1, PQ2} is submitted for certification, it SHOULD be rejected because the key RSA1 was previously declared compromised even though the key PQ2 is unique.

6. Appendices

6.1. ASN.1 Module

```
<CODE STARTS>

Composite-Signatures-2019
  { TBD }

DEFINITIONS IMPLICIT TAGS ::= BEGIN

EXPORTS ALL;

IMPORTS
  PUBLIC-KEY, SIGNATURE-ALGORITHM
    FROM AlgorithmInformation-2009  -- RFC 5912 [X509ASN1]
      { iso(1) identified-organization(3) dod(6) internet(1)
        security(5) mechanisms(5) pkix(7) id-mod(0)
        id-mod-algorithmInformation-02(58) }

  SubjectPublicKeyInfo
    FROM PKIX1Explicit-2009
      { iso(1) identified-organization(3) dod(6) internet(1)
        security(5) mechanisms(5) pkix(7) id-mod(0)
        id-mod-pkix1-explicit-02(51) }

  OneAsymmetricKey
    FROM AsymmetricKeyPackageModuleV1
      { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
        pkcs-9(9) smime(16) modules(0)
        id-mod-asymmetricKeyPkgV1(50) } ;

--
-- Object Identifiers
--

id-alg-composite OBJECT IDENTIFIER ::= { TBD }

--
-- Public Key
--

pk-Composite PUBLIC-KEY ::= {
    IDENTIFIER id-alg-composite
    KEY CompositePublicKey
    PARAMS ARE absent
    PRIVATE-KEY CompositePrivateKey
}

CompositePublicKey ::= SEQUENCE SIZE (2..MAX) OF SubjectPublicKeyInfo

CompositePrivateKey ::= SEQUENCE SIZE (2..MAX) OF OneAsymmetricKey
```

```
END
```

<CODE ENDS>

6.2. Intellectual Property Considerations

The following IPR Disclosure relates to this draft:

https://datatracker.ietf.org/ipr/3588/

7. Contributors and Acknowledgements

This document incorporates contributions and comments from a large group of experts. The Editors would especially like to acknowledge the expertise and tireless dedication of the following people, who attended many long meetings and generated millions of bytes of electronic mail and VOIP traffic over the past year in pursuit of this document:

John Gray (Entrust), Serge Mister (Entrust), Scott Fluhrer (Cisco Systems), Panos Kampanakis (Cisco Systems), Daniel Van Geest (ISARA), Tim Hollebeek (Digicert), and Francois Rousseau.

We are grateful to all, including any contributors who may have been inadvertently omitted from this list.

This document borrows text from similar documents, including those referenced below. Thanks go to the authors of those documents. "Copying always makes things easier and less error prone" - [RFC8411].

7.1. Making contributions

Additional contributions to this draft are weclome. Please see the working copy of this draft at, as well as open issues at:

https://github.com/EntrustCorporation/draft-ounsworth-pq-composite-keys

8. Normative References

[RFC1421] Linn, J., "Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures", RFC 1421, DOI 10.17487/RFC1421, February 1993, <https://www.rfc-editor.org/info/rfc1421>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, DOI

10.17487/RFC2986, November 2000, <https://www.rfc-editor.org/info/rfc2986>.

[RFC4210] Adams, C., Farrell, S., Kause, T., and T. Mononen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)", RFC 4210, DOI 10.17487/RFC4210, September 2005, <https://www.rfc-editor.org/info/rfc4210>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/info/rfc5652>.

[RFC5914] Housley, R., Ashmore, S., and C. Wallace, "Trust Anchor Format", RFC 5914, DOI 10.17487/RFC5914, June 2010, <https://www.rfc-editor.org/info/rfc5914>.

[RFC5958] Turner, S., "Asymmetric Key Packages", RFC 5958, DOI 10.17487/RFC5958, August 2010, <https://www.rfc-editor.org/info/rfc5958>.

[RFC7468] Josefsson, S. and S. Leonard, "Textual Encodings of PKIX, PKCS, and CMS Structures", RFC 7468, DOI 10.17487/RFC7468, April 2015, <https://www.rfc-editor.org/info/rfc7468>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8411] Schaad, J. and R. Andrews, "IANA Registration for the Cryptographic Algorithm Object Identifier Range", RFC 8411, DOI 10.17487/RFC8411, August 2018, <https://www.rfc-editor.org/info/rfc8411>.

[X.690] ITU-T, "Information technology - ASN.1 encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ISO/IEC 8825-1:2015, November 2015.

Authors' Addresses

Mike Ounsworth
Entrust Limited

2500 Solandt Road -- Suite 100
Ottawa, Ontario K2K 3G5
Canada

Email: mike.ounsworth@entrust.com

Massimiliano Pala
CableLabs

Email: director@openca.org