Workgroup: LAMPS
Internet-Draft:
draft-ounsworth-pq-composite-keys-03
Published: 22 October 2022
Intended Status: Standards Track
Expires: 25 April 2023
Authors: M. Ounsworth M. Pala J. Klaussner
Entrust CableLabs D-Trust GmbH

Composite Public and Private Keys For Use In Internet PKI

Abstract

The migration to post-quantum cryptography is unique in the history of modern digital cryptography in that neither the old outgoing nor the new incoming algorithms are fully trusted to protect data for the required data lifetimes. The outgoing algorithms, such as RSA and elliptic curve, may fall to quantum cryptalanysis, while the incoming post-quantum algorithms face uncertainty about both the underlying mathematics as well as hardware and software implementations that have not had sufficient maturing time to rule out classical cryptanalytic attacks and implementation bugs.

Cautious implementors may wish to layer cryptographic algorithms such that an attacker would need to break all of them in order to compromise the data being protected using either a Post-Quantum / Traditional Hybrid, Post-Quantum / Post-Quantum Hybrid, or combinations thereof. This document, and its companions, defines a specific instantiation of hybrid paradigm called "composite" where multiple cryptographic algorithms are combined to form a single key, signature, or key encapsulation mechanism (KEM) such that they can be treated as a single atomic object at the protocol level.

This document defines the structures CompositePublicKey and CompositePrivateKey, which are sequences of the respective structure for each component algorithm. The generic composite variant is defined which allows arbitrary combinations of key types to be placed in the CompositePublicKey and CompositePrivateKey structures without needing the combination to be pre-registered or pre-agreed. The explicit variant is alxso defined which allows for a set of algorithm identifier OIDs to be registered together as an explicit composite algorithm and assigned an OID.

This document is intended to be coupled with corresponding documents that define the structure and semantics of composite signatures and encryption, such as [I-D.ounsworth-pq-composite-sigs] and [I-D.ounsworth-pq-composite-kem].

Status of This Memo

Copyright Notice

Table of Contents

1. Changes in version -03

*Added the following explicit composite key types

-Explicit Composite Signature Keys

oid-Dilithium3-ECDSA-P256

oid-Dilithium3-RSA

oid-Falcon512-ECDSA-P256

oid-Falcon512-Ed25519

oid-SPHINCSXXX-ECDSA-P256

oid-Dilithium5-Falcon1024-ECDSA-P521

oid-Dilithium5-Falcon1024-RSA

-Explicit Composite KEM Keys

oid-Kyber512-RSA

oid-Kyber512-ECDH-P256

oid-Kyber512-x25519

*Added samples of (most of) the above explic composites in appendices.

*Marked generic composite for prototyping; to be removed in final publication.

*Sycronized terminology with I-D.draft-driscoll-pqt-hybrid-terminology-01.

*Removed the section "Implementation Considerations > Asymmetric Key Packages (CMS)" since private key formats are now fully covered in the body and examples.

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.

This document is consistent with all terminology from [I-D.driscoll-pqt-hybrid-terminology].

In addition, the following terms are used in this document:

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

CLIENT: Any software that is making use of a cryptographic key. This includes a signer, verifier, encrypter, decrypter.

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

PKI: Public Key Infrastructure, as defined in [RFC5280].

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

2. Introduction

During the transition to post-quantum cryptography (PQ or PQC), 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 may also not fully trust their post-quantum replacements until further time has passed to allow additional scrutiny and the discovery of implementation bugs. 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 by using a Post-Quantum/Traditional (PQ/T) or Post-Quantum/Post-Quantum (PQ/PQ) Hybrid scheme.

The transition to PQC 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.

* Migration: During the transition period, systems will require mechanisms that allow for staged migrations from fully traditional to fully post-quantum-aware cryptography.

This document provides the composite mechanism, which is a specific instantiation of the PQ/T and PQ/PQ Hybrid paradigm to address algorithm strength uncertainty concerns by providing formats for encoding multiple public key and private key values into existing public key and private key fields. Backwards compatibility is not directly addressed via the composite mechanisms defined in the document, but some notes on how it can be obtained can be found in Section 5.2.

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

3. Composite Key Structures

In order to represent public keys and private keys that are composed of multiple algorithms, we define encodings consisting of a sequence of public key or private key primitives (aka "components") such that these structures can be used directly in 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].

A composite key is a single key object that performs an atomic cryptographic operation -- such a signing, verifying, encapsulating, or decapsulating -- using its encapsulated sequence of component keys as if it was a single key. This generally means that the complexity of combining algorithms can be deferred from the protocol layer to the cryptographic library layer.

3.1. CompositePublicKey

Composite public key data is represented by the following structure:

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

A composite key MUST contain at least two component public keys.

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 and therefore reduction in attack surface.

Each component SubjectPublicKeyInfo SHALL contain an AlgorithmIdentifier OID which identifies the public key type and parameters for the public key contained within it. See Section 4 for specific algorithms defined in this document.

Each element of a CompositePublicKey is a SubjectPublicKeyInfo object encoding 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 3.3.

3.2. CompositePrivateKey

This section provides an encoding for composite private keys intended for PKIX protocols and other applications that require an interoperable format for transmitting private keys, such as PKCS #12 [RFC7292] or CMP / CRMF [RFC4210], [RFC4211]. It is not intended to

dictate a storage format in implementations not requiring interoperability of private key formats.

In some cases the private keys that comprise a composite key may not be represented in a single structure or even be contained in a single cryptographic module. The establishment of correspondence between public keys in a CompositePublicKey and private keys not represented in a single composite structure is beyond the scope of this document.

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 parameters field MUST be absent.

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

EDNOTE: does this also need an explicit version? It would probably reduce attack surface of tricking a client into running the wrong parser and a given piece of data. ... maybe we get that for free if we use the explicit composite OIDs also for private keys?

3.2.1. 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. The composite mechanism 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.

3.3. 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.

```
CompositePublicKeyOs ::= OCTET STRING (CONTAINING CompositePublicKey ENCODED BY der)
```

EDNOTE: will this definition include an ASN.1 tag and length byte inside the OCTET STRING object? If so, that's probably an extra uneccessary layer.

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.

```
CompositePublicKeyBs ::= BIT STRING (CONTAINING CompositePublicKey ENCODED BY der)
```

EDNOTE: See this LAMPS mailist discussion about BIT STRING vs OCTET STRING for public keys. I think we have dodged the issue, but may we worth re-visiting. https://mailarchive.ietf.org/arch/msg/spasm/Gv-ACiOpYZfOM0AJEZUX1jIhVq0/

4. Algorithm Identifiers

This section defines the algorithm identifier for generic composite, as well as a framework for defining explicit combinations and a list of explicit composite algorithms covering a wide range of use cases. This section is not intended to be exhaustive and other authors may define others so long as they are compatible with the structures and processes defined in this and companion signature and encryption documents.

Some use-cases desire the flexibility for client to use any combination of supported algorithms, while others desire the rigidity of explicitly-specified combinations of algorithms.

4.1. id-composite-key (Generic Composite Keys)

Usage guidance: This mode is primarily for prototyping and for use in proprietary implementations; implementers MAY implement this section if there is a need for greater flexibility in algorithm combinations than is available by using the pre-defined composite algorithms defined below.

EDNOTE: Does the WG feel strongly that this section should be removed prior to publication by the RFC Editors? IE remove it

entirely from the standard? Are there enduring (ie non-prototyping) usecases that benefit from generic composite?

The id-composite-key object identifier is used for identifying a generic composite public key and a generic composite private key. This allows arbitrary combinations of key types to be placed in the CompositePublicKey and CompositePrivateKey structures without needing the combination to be pre-registered or standardized.

```
id-composite-key OBJECT IDENTIFIER ::= {
    joint-iso-itu-t(2) country(16) us(840) organization(1) entrust(114027)
    Algorithm(80) Composite(4) CompositeKey(1) }
```

EDNOTE: this is a temporary OID for the purposes of prototyping. We are requesting IANA to assign a permanent OID, see Section 6.

The PUBLIC-KEY ASN.1 information object class is defined in [RFC5912]. The PUBLIC-KEY information object for generic (Section 4.1) and explicit (Section 4.2) composite public and private keys has the following form:

```
pk-Composite PUBLIC-KEY ::= {
    id id-composite-key
    KeyValue CompositePublicKey
    Params ARE ABSENT
    PrivateKey CompositePrivateKey
}
```

The motivation for this variant is primarily for prototyping work prior to the standardization of algorithm identifiers for explicit combinations of algorithms. However, the authors envision that this variant will remain relevant beyond full standardization for example in environments requiring very high levels of crypto agility, for example where clients support a large number of algorithms or where a large number of keys will be used at a time and it is therefore prohibitive to define algorithm identifiers for every combination of pairs, triples, quadruples, etc of algorithms.

4.2. Explicit Composite Signature Keys

This variant provides a rigid way of specifying supported combinations of key types.

The motivation for this variant is to make it easier to reference and enforce specific combinations of algorithms. The authors envision this being useful for client-server negotiated protocols,

protocol designers who wish to place constraints on allowable algorithm combinations in the protocol specification, as well as audited environments that wish to prove that only certain combinations will be supported by clients.

Profiles need to define an explicit composite key type which consists of, at the minimum:

- A new algorithm identifier OID for the explicit algorithm.
- The PUBLIC-KEY information object of each component public key type.
- The algorithm identifiers and parameters to be contained in each of the component SubjectPublicKeyInfos and OneAsymmetricKeys.

See Appendix A for guidance on creating and registering OIDs for specific explicit combinations.

In this variant, the public key is encoded as defined in Section 3 and Section 3.1, however the PUBLIC-KEY.id SHALL be an OID which is registered to represent a specific combination of component public key types. See Appendix B for examples.

The SubjectPublicKeyInfo.algorithm for each component key is redundant information which MUST match -- and can be inferred from -- the specification of the explicit algorithm. It has been left here for ease of implementation as the component SubjectPublicKeyInfo structures are the same between generic and explicit, as well as with single-algorithm keys. However, it introduces the risk of mismatch and leads to the following security consideration:

Security consideration: Implementations MUST check that the component AlgorithmIdentifier OIDs and parameters match those expected by the definition of the explicit algorithm. Implementations SHOULD first parse a component's SubjectPublicKeyInfo.algorithm, and ensure that it matches what is expected for that position in the explicit key, and then proceed to parse the SubjectPublicKeyInfo.subjectPublicKey. This is to reduce the attack surface associated with parsing the public key data of an unexpected key type, or worse; to parse and use a key which does not match the explicit algorithm definition. Similar checks MUST be done when handling the corresponding private key.

Below are provided a set of explicit composite algorithms which have been selected to fill a wide range of use cases, with algorithms selected to match security levels across a group. The selections include pairs of {lattice, ECC/RSA} which should cover most short-term use cases, and also {hash-based, ECC} pairs and {lattice,

lattice, ECC/RSA} triples for long-term use cases. Usage guidance is provided for each explicit combination.

The algorithm set provided here is not intended to be exhaustive; additional use cases and cryptographic advances may require additional combinations to be defined in other documents by using the mechanim provided in Appendix A.

4.2.1. id-Dilithium3-ECDSA-P256

Usage guidance: This signature key type is intended to be the standard composite signature, applicable for most use cases.

The following object identifier is defined:

```
id-Dilithium3-ECDSA-P256 OBJECT IDENTIFIER ::= {
  joint-iso-itu-t(2) country(16) us(840) organization(1) entrust(114027)
  algorithm(80) ExplicitCompositeKey(5) id-Dilithium3-ECDSA-P256(1) }
```

EDNOTE: this is a temporary OID for the purposes of prototyping. We are requesting IANA to assign a permanent OID, see Section 6.

When used in an AlgorithmIdentifier, parameters SHALL be ABSENT.

The PUBLIC-KEY SHALL be:

```
pk-Dilithium3-ECDSA-P256 PUBLIC-KEY ::= {
    id id-Dilithium3-ECDSA-P256
    KeyValue CompositePublicKey
    Params ARE ABSENT
    PrivateKey CompositePrivateKey
}
```

--- BEGIN EDNOTE ---

EDNOTE: design decision needed: should it be CompositePublicKey and CompositePrivateKey, or should we define Dilithium3-ECDSA-P256-PublicKey and Dilithium3-ECDSA-P256-PrivateKey for each explicit type?

Pros of a *-PublicKey for each explicit composite type: 1) the ASN.1 parser will do some of the heavy-lifting of checking that types match (but how much is up for debate..). 2) can further compress the encoding by removing the redundant inner component OIDs.

Pros of CompositePublicKey (ie carrying full SPKIs for each component): 1) it becomes harder to write abstract code that takes advantange of the fact that all composite explicit types have the same wire encoding structure because each will have an independantly defined structure object. 2) The wire encoding carries a full SPKI, so for crypto libraries that require an SPKI for each component alg, clients need to reconstruct a full SPKI, including reconstituting the components OIDs, which needs the client to have a mapping table of explicit composite OIDs to component OIDs.

--- END EDNOTE ---

The public key encoding is defined in Section 3.1 for id-Dilithium3-ECDSA-P256 SHALL have SIZE (2):

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

The first SubjectPublicKeyInfo (defined in [RFC5280]) SHALL contain:

```
SEQUENCE  {
     algorithm  AlgorithmIdentifier {
          algorithm id-dilithiumTBD,
          paramaters ABSENT
      },
     subjectPublicKey     BIT STRING(DilithiumPublicKey)
}
```

where pk-dilithiumTBD and TBDDilithiumPublicKey are defined in [I-D.massimo-lamps-pq-sig-certificates].

TODO: I don't think subjectPublicKey BIT STRING(DilithiumPublicKey) is correct ASN.1.

EDNOTE: pk-dilithiumTBD and TBDPublicKey refer to [I-D.massimo-lamps-pq-sig-certificates] and should be kept in sync with future versions of that draft.

The second SubjectPublicKeyInfo SHALL contain:

```
SEQUENCE  {
     algorithm  AlgorithmIdentifier {
            algorithm id-ecPublicKey,
            parameters secp256r1
      },
     subjectPublicKey     BIT STRING(ECPoint)
}
```

where id-ecPublicKey, secp256r1, and ECPoint are defined in [RFC5912].

The private key encoding is defined in Section 3.2, and for id-Dilithium3-ECDSA-P256 SHALL have SIZE (2):

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

The first OneAsymmetricKey, defined in [RFC5958] SHALL contain

```
privateKeyAlgorithm AlgorithmIdentifier ::= {
    algorithm id-dilithiumTBD,
    parameters ABSENT
  },
privateKey DilithiumPrivateKey
```

where id-dilithiumTBD and DilithiumPrivateKey are defined in [I-D.massimo-lamps-pq-sig-certificates].

The publicKey remains OPTIONAL.

The second OneAsymmetricKey SHALL contain

```
privateKeyAlgorithm AlgorithmIdentifier ::= {
    algorithm id-ecPublicKey,
    parameters secp256r1
  },
privateKey ECPrivateKey
```

where ECPrivateKey is defined in [RFC5480].

The publicKey remains OPTIONAL.

4.2.2. id-Dilithium3-RSA

Usage guidance: This signature key type is intended to be the standard composite signature for environments that support RSA but not elliptic curve.

The following object identifier is defined:

```
id-Dilithium3-RSA OBJECT IDENTIFIER ::= {
  joint-iso-itu-t(2) country(16) us(840) organization(1) entrust(114027)
  algorithm(80) ExplicitCompositeKey(5) id-Dilithium3-RSA(2) }
```

EDNOTE: this is a temporary OID for the purposes of prototyping. We are requesting IANA to assign a permanent OID, see Section 6.

When used in an AlgorithmIdentifier, parameters SHALL be ABSENT.

The PUBLIC-KEY SHALL be:

```
pk-Dilithium3-ECDSA-P256 PUBLIC-KEY ::= {
    id id-Dilithium3-RSA
    KeyValue CompositePublicKey
    Params ARE ABSENT
    PrivateKey CompositePrivateKey
}
```

--- BEGIN EDNOTE ---

EDNOTE: design decision needed: should it be CompositePublicKey + CompositePrivateKey, or should we define *-PublicKey and *-PrivateKey for each explicit type?

Pros of a *-PublicKey for each explicit composite type: 1) the ASN.1 parser will do some of the heavy-lifting of checking that types match (but how much is up for debate..). 2) can further compress the encoding by removing the redundant inner component OIDs.

Pros of CompositePublicKey (ie carrying full SPKIs for each component): 1) it becomes harder to write abstract code that takes advantange of the fact that all composite explicit types have the same wire encoding structure because each will have an independantly defined structure object. 2) The wire encoding carries a full SPKI, so for crypto libraries that require an SPKI for each component alg, clients need to reconstruct a full SPKI, including reconstituting the components OIDs, which needs the client to have a mapping table of explicit composite OIDs to component OIDs.

--- END EDNOTE ---

The public key encoding is defined in Section 3.1 for id-Dilithium3-ECDSA-P256 SHALL have SIZE (2):

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

The first SubjectPublicKeyInfo (defined in [RFC5280]) SHALL contain:

```
SEQUENCE  {
     algorithm  AlgorithmIdentifier {
          algorithm id-dilithiumTBD,
          paramaters ABSENT
      },
     subjectPublicKey     BIT STRING(DilithiumPublicKey)
}
```

where pk-dilithiumTBD and TBDDilithiumPublicKey are defined in [I-D.massimo-lamps-pq-sig-certificates].

TODO: I don't think subjectPublicKey BIT STRING(DilithiumPublicKey) is correct ASN.1.

EDNOTE: pk-dilithiumTBD and TBDPublicKey refer to [I-D.massimo-lamps-pq-sig-certificates] and should be kept in sync with future versions of that draft.

The second SubjectPublicKeyInfo SHALL contain:

```
SEQUENCE  {
     algorithm  AlgorithmIdentifier {
            algorithm rsaEncryption,
            parameters NULL
      },
     subjectPublicKey     BIT STRING(RSAPublicKey)
}
```

where rsaEncryption and RSAPublicKey are defined in [RFC8017].

The private key encoding is defined in Section 3.2, and for id-Dilithium3-ECDSA-P256 SHALL have SIZE (2):

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

The first OneAsymmetricKey, defined in [RFC5958] SHALL contain

```
privateKeyAlgorithm AlgorithmIdentifier ::= {
    algorithm id-dilithiumTBD,
    parameters ABSENT
  },
privateKey DilithiumPrivateKey
```

The publicKey remains OPTIONAL.

The second OneAsymmetricKey SHALL contain

```
privateKeyAlgorithm AlgorithmIdentifier ::= {
    algorithm id-ecPublicKey,
    parameters secp256r1
  },
privateKey ECPrivateKey
```

where ECPrivateKey is defined in [RFC5480].

The publicKey remains OPTIONAL.

4.2.3. id-Falcon512-ECDSA-P256

Usage guidance: This signature key type is intended for constrained environments that need to use FIPS-approved elliptic curve.

TODO: fill in details.

4.2.4. id-Falcon512-Ed25519

Usage guidance: This signature key type is intended for constrained environments that may use non-FIPS-approved elliptic curve.

TODO: fill in details.

4.2.5. id-SPHINCSsha256256frobust-ECDSA-P256

Usage guidance: This signature key type is intended for long-term keys that desire the robustness to algorithmic attacks that comes from stateless hash-based signatures as well as the robustness to implementation bugs that comes from coupling with mature ECDSA implementations.

TODO: fill in details.

4.2.6. id-Dilithium5-Falcon1024-ECDSA-P521

Usage guidance: This signature key type is intended for long-term keys that desire robustness to the break of a given lattice-based scheme, but also desire smaller signatures than stateless hash-based signatures.

Note that this still has smaller pubkey + sig than SPHINCS+. TODO: fill in numbers.

TODO: fill in details.

4.2.7. id-Dilithium5-Falcon1024-RSA

Usage guidance: This signature key type is intended for long-term keys that desire robustness to the break of a given lattice-based

scheme for environments that support RSA but not elliptic curve, but also desire smaller signatures than stateless hash-based signatures.

Note that this still has smaller pubkey + sig than SPHINCS+. TODO: fill in numbers.

TODO: fill in details.

4.3. Explicit Composite KEM Keys

TODO: some text

Ref to draft-ounsworth-pq-composite-kem

4.3.1. id-Kyber512-RSA

TODO

4.3.2. id-Kyber512-ECDH-P256

TODO

4.3.3. id-Kyber512-x25519

TODO

5. Implementation Considerations

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

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.

5.1. 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.

5.2. Backwards Compatibility

As noted in the introduction, the post-quantum cryptographic migration will face challenges in both ensuring cryptographic strength against adversaries of unknown capabilities, as well as providing ease of migration. The composite mechanisms defined in this document primarily address cryptographic strength, however this section contains notes on how backwards compatibility may be obtained.

The term "ease of migration" is used here to mean that existing systems can be gracefully transitioned to the new technology without requiring large service disruptions or expensive upgrades. The term "backwards compatibility" is used here to mean something more specific; that existing systems, as they are deployed today, can interoperate with the upgraded systems of the future.

These migration and interoperability concerns need to be thought about in the context of various types of protocols that make use of X.509 and PKIX with relation to public key objects, from online negotiated protocols such as TLS 1.3 [RFC8446] and IKEv2 [RFC7296], to non-negotiated asynchronous protocols such as S/MIME signed and encrypted email [RFC8551], document signing such as in the context of the European eIDAS regulations [eIDAS2014], and publicly trusted code signing [codeSigningBRsv2.8], as well as myriad other standardized and proprietary protocols and applications that leverage CMS [RFC5652] signed or encrypted structures.

5.2.1. OR modes

This document purposefully does not specify how clients are to combine component keys together to form a single cryptographic operation; this is left up to the specifications of signature and encryption algorithms that make use of the composite key type. One possible way to combine component keys is through an OR relation, or OR-like client policies for acceptable algorithm combinations, where senders and / or receivers are permitted to ignore some component keys. Some envisioned uses of this include environments where the client encounters a component key for which it does not possess a compatible algorithm implementation but wishes to proceed with the cryptographic operation using the subset of component keys for which it does have compatible implementations. Such a mechanism could be designed to provide ease of migration by allowing for composite keys to be distributed and used before all clients in the environment are fully upgraded, but it does not allow for full backwards

compatibility since clients would at least need to be upgraded from their current state to be able to parse the composite structures.

5.2.2. Parallel PKIs

We present the term "Parallel PKI" to refer to the setup where a PKI end entity possesses two or more distinct public keys or certificates for the same key type (signature, key establishment, etc) for the same identity (name, SAN), but containing keys for different cryptographic algorithms. One could imagine a set of parallel PKIs where an existing PKI using legacy algorithms (RSA, ECC) is left operational during the post-quantum migration but is shadowed by one or more parallel PKIs using pure post quantum algorithms or composite algorithms (legacy and post-quantum). This concept contains strong overlap with other documented approaches, such as [I-D.becker-guthrie-noncomposite-hybrid-auth] and highlights the synergy between composite and non-composite hybrid approaches.

Equipped with a set of parallel public keys in this way, a client would have the flexibility to choose which public key(s) or certificate(s) to use in a given cryptographic operation.

For negotiated protocols, the client could choose which public key(s) or certificate(s) to use based on the negotiated algorithms, or could combine two of the public keys for example in a non-composite hybrid method such as [I-D.becker-guthrie-noncomposite-hybrid-auth] or [I-D.guthrie-ipsecme-ikev2-hybrid-auth]. Note that it is possible to use the signature algorithm defined in [I-D.ounsworth-pq-composite-sigs] as a way to carry the multiple signature values generated by a non-composite public mechanism in protocols where it is easier to support the composite signature algorithms than to implement such a mechanism in the protocol itself. There is also nothing precluding a composite public key from being one of the components used within a non-composite authentication operation; this may lead to greater convenience in setting up parallel PKI hierarchies that need to service a range of clients implementing different styles of post-quantum migration strategies.

For non-negotiated protocols, the details for obtaining backwards compatibility will vary by protocol, but for example in CMS [RFC5652], the inclusion of multiple SignerInfo or RecipientInfo objects is often already treated as an OR relationship, so including one for each of the end entity's parallel PKI public keys would, in many cases, have the desired effect of allowing the receiver to choose one they are compatible with and ignore the others, thus achieving full backwards compatibility.

6. IANA Considerations

This document registers the following in the SMI "Security for PKIX Algorithms (1.3.6.1.5.5.7.6)" registry:

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

7. Security Considerations

7.1. 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 outside of their composite context and therefore it is RECOMMENDED that component keys in a composite key are not to be re-used in other contexts. In particular, the components of a composite key SHOULD NOT also appear in single-key certificates. This is particularly relevant for protocols that use composite keys in a logical AND mode since the appearance of the same component keys in single-key contexts undermines the binding of the component keys into a single composite key by allowing messages signed in a multi-key AND mode to be presented as if they were signed in a single key mode in what is known as a "stripping attack".

7.2. Key mismatch in explicit composite

This security consideration copied from Section 4.2.

Implementations MUST check that that the component AlgorithmIdentifier OIDs and parameters match those expected by the definition of the explicit algorithm. Implementations SHOULD first parse a component's SubjectPublicKeyInfo.algorithm, and ensure that it matches what is expected for that position in the explicit key, and then proceed to parse the SubjectPublicKeyInfo.subjectPublicKey. This is to reduce the attack surface associated with parsing the public key data of an unexpected key type, or worse; to parse and use a key which does not match the explicit algorithm definition. Similar checks MUST be done when handling the corresponding private key.

7.3. 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

clients performing signature verification or encryption operations should be updated to fail to validate or refuse to encrypt for these algorithms.

In the composite model this is less obvious since implementers may decide that certain cryptographic algorithms have complementary security properties and are acceptable in combination even though one or both algorithms are deprecated for individual use. As such, a single composite public key, certificate, signature, or ciphertext MAY contain a mixture of deprecated and non-deprecated algorithms.

Specifying behaviour in these cases is beyond the scope of this document, but should be considered by implementers and potentially in additional standards.

EDNOTE: Max had proposed a CRL mechanism to accomplish this, which could be revived if necessary.

7.4. 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.

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.

7.5. Checking for Compromised Key Reuse

Certification 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 individually 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.

8. References

8.1. Normative References

[I-D.massimo-lamps-pq-sig-certificates] Massimo, J., Kampanakis, P., Turner, S., and B. Westerbaan, "Algorithms and Identifiers for Post-Quantum Algorithms", Work in Progress, Internet-Draft, draft-massimo-lamps-pq-sig-certificates-00, 8 July 2022, <https://www.ietf.org/archive/id/draft-massimo-lamps-pq-sig-certificates-00.txt>.

[I-D.ounsworth-pq-composite-kem] Ounsworth, M. and J. Gray, "Composite KEM For Use In Internet PKI", Work in Progress, Internet-Draft, draft-ounsworth-pq-composite-kem-00, 11 July 2022, <https://www.ietf.org/archive/id/draft-ounsworth-pq-composite-kem-00.txt>.

[I-D.ounsworth-pq-composite-sigs] Ounsworth, M. and M. Pala, "Composite Signatures For Use In Internet PKI", Work in Progress, Internet-Draft, draft-ounsworth-pq-composite-sigs-05, 12 July 2021, <https://www.ietf.org/archive/id/draft-ounsworth-pq-composite-sigs-05.txt>.

[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>.

[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>.

[RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", RFC 5480, DOI 10.17487/RFC5480, March 2009, <https://www.rfc-editor.org/info/rfc5480>.

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

[RFC5912] Hoffman, P. and J. Schaad, "New ASN.1 Modules for the Public Key Infrastructure Using X.509 (PKIX)", RFC 5912, DOI 10.17487/RFC5912, June 2010, <https://www.rfc-editor.org/info/rfc5912>.

[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>.

[RFC5915] Turner, S. and D. Brown, "Elliptic Curve Private Key Structure", RFC 5915, DOI 10.17487/RFC5915, June 2010, <https://www.rfc-editor.org/info/rfc5915>.

[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>.

[RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <https://www.rfc-editor.org/info/rfc8017>.

[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.

8.2. Informative References

[Beullens2022] Beullens, W., "Breaking rainbow takes a weekend on a laptop.", 2022, <https://eprint.iacr.org/2022/214>.

[Bindel2017]
Bindel, N., Herath, U., McKague, M., and D. Stebila, "Transitioning to a quantum-resistant public key infrastructure", 2017, <https://link.springer.com/chapter/10.1007/978-3-319-59879-6_22>.

[Bleichenbacher1998] Bleichenbacher, D., "Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS# 1.", 1998.

[Castryck2022] Castryck, W. and T. Decru, "An efficient key recovery attack on SIDH (preliminary version).", 2022, <https://eprint.iacr.org/2022/975.pdf>.

[codeSigningBRsv2.8] CAB Forum, "Baseline Requirements for the Issuance and Management of Publicly-Trusted Code Signing Certificates v2.8", May 2022, <https://cabforum.org/wp-content/uploads/Baseline-Requirements-for-the-Issuance-and-Management-of-Code-Signing.v2.8.pdf>.

[eIDAS2014] "REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC", July 2014, <https://ec.europa.eu/futurium/en/system/files/ged/eidas_regulation.pdf>.

[I-D.becker-guthrie-noncomposite-hybrid-auth] Becker, A., Guthrie, R., and M. Jenkins, "Non-Composite Hybrid Authentication in PKIX and Applications to Internet Protocols", Work in Progress, Internet-Draft, draft-becker-guthrie-noncomposite-hybrid-auth-00, 22 March 2022, <https://www.ietf.org/archive/id/draft-becker-guthrie-noncomposite-hybrid-auth-00.txt>.

[I-D.driscoll-pqt-hybrid-terminology]
D, F., "Terminology for Post-Quantum Traditional Hybrid Schemes", Work in Progress, Internet-Draft, draft-driscoll-pqt-hybrid-terminology-01, 20 October 2022, <https://www.ietf.org/archive/id/draft-driscoll-pqt-hybrid-terminology-01.txt>.

[I-D.guthrie-ipsecme-ikev2-hybrid-auth]
Guthrie, R., "Hybrid Non-Composite Authentication in IKEv2", Work in Progress, Internet-Draft, draft-guthrie-ipsecme-ikev2-hybrid-auth-00, 25 March 2022, <https://

www.ietf.org/archive/id/draft-guthrie-ipsecme-ikev2-hybrid-auth-00.txt>.

[Mosca2015] Mosca, M., "Cybersecurity in a Quantum World: will we be ready?", April 2015, <https://csrc.nist.gov/csrc/media/events/workshop-on-cybersecurity-in-a-post-quantum-world/documents/presentations/session8-mosca-michele.pdf>.

[RFC3279] Bassham, L., Polk, W., and R. Housley, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3279, DOI 10.17487/RFC3279, April 2002, <https://www.rfc-editor.org/info/rfc3279>.

[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>.

[RFC4211] Schaad, J., "Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)", RFC 4211, DOI 10.17487/RFC4211, September 2005, <https://www.rfc-editor.org/info/rfc4211>.

[RFC7292] Moriarty, K., Ed., Nystrom, M., Parkinson, S., Rusch, A., and M. Scott, "PKCS #12: Personal Information Exchange Syntax v1.1", RFC 7292, DOI 10.17487/RFC7292, July 2014, <https://www.rfc-editor.org/info/rfc7292>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8551] Schaad, J., Ramsdell, B., and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification", RFC 8551, DOI 10.17487/RFC8551, April 2019, <https://www.rfc-editor.org/info/rfc8551>.

[ROBOT2018] Boeck, H., Somorovsky, J., and C. Young, "Return Of {Bleichenbacher's} Oracle Threat (ROBOT).", 2018, <https://www.usenix.org/conference/usenixsecurity18/presentation/bock>.

Appendix A. Creating explicit combinations

The following ASN.1 Information Objects may be useful in defining and parsing explicit pairs of public key types. Given an ASN.1 2002 compliant ASN.1 compiler, these Information Objects will enforce the binding between the public key types specified in the instantiation of pk-explicitComposite, and the wire objects which implement it. The one thing that is not enforced automatically by this Information Object is that publicKey.params are intended to be absent if and only if they are absent for the declared public key type. This ASN.1 module declares them OPTIONAL and leaves it to implementers to perform this check explicitly.

EDNOTE this ASN.1 needs to change. The current definition doesn't put a component AlgorithmIdentifier with each component key. Once we agree as a group that the text accurately describes what we want, we can spend a bit of time figuring out if the ASN.1 machinery lets us express it in a readable way and/or a way that will actually help people creating explicit pairs.

```
-- pk-explicitComposite - Composite public key information object

pk-explicitComposite{OBJECT IDENTIFIER:id, PUBLIC-KEY:firstPublicKey,
 FirstPublicKeyType, PUBLIC-KEY:secondPublicKey, SecondPublicKeyType}
 PUBLIC-KEY ::= {PUBLIC-KEYPUBLIC-KEY
    IDENTIFIER id
    KEY ExplicitCompositePublicKey{firstPublicKey, FirstPublicKeyType,
     secondPublicKey, SecondPublicKeyType}
    PARAMS ARE absent
    CERT-KEY-USAGE {digitalSignature, nonRepudiation, keyCertSign,
      cRLSign}
}
```

The following ASN.1 object class then automatically generates the public key structure from the types defined in pk-explicitComposite.

```
-- ExplicitCompositePublicKey - The data structure for a composite
-- public key sec-composite-pub-keys and SecondPublicKeyType are needed
-- because PUBLIC-KEY contains a set of public key types, not a single
-- type.
-- TODO The parameters should be optional only if they are marked
-- optional in the PUBLIC-KEY.


ExplicitCompositePublicKey{PUBLIC-KEY:firstPublicKey, FirstPublicKeyType,
  PUBLIC-KEY:secondPublicKey, SecondPublicKeyType} ::= SEQUENCE {
    firstPublicKey SEQUENCE {
        params firstPublicKey.&Params OPTIONAL,
        publicKey FirstPublicKeyType
    },
    secondPublicKey SEQUENCE {
        params secondPublicKey.&Params OPTIONAL,
        publicKey SecondPublicKeyType
    }
}
```

Using this module, it becomes trivial to define explicit pairs. For an example, see Appendix B.2.

To define explicit triples, quadruples, etc, these Information Objects can be extended to have thirdPublicKey, fourthPublicKey, etc throughout.

Appendix B. Examples

These samples are reproduced here for completeness, but are also available in github:

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

B.1. Generic Composite Public Key Examples

This is an example generic composite public key

```
-----BEGIN PUBLIC KEY-----
MIIBmDAMBgpghkgBhvprUAQBA4IBhgAwggGBMFkwEwYHKoZIzj0CAQYIKoZIzj0D
AQcDQgAExGPhrnuSG/fGyw1FN+l5h4p4AGRQCS0LBXnBO+djhcI6qnF2TvrQEaIY
GGpQT5wHS+7y5iJJ+dE5qjxcv8loRDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCC
AQoCggEBANsVQK1fcLQObL4ZYtczWbObECAFSsng0OLpRTPr9VGV3SsS/VoMRZqX
F+sszz6I2UcFTaMF9CwNRbWLuIBczzuhbHSjn65OuoN+Om2wsPo+okw46RTekB4a
d9QQvYRVzPlILUQ8NvZ4W0BKLviXTXWIggjtp/Y1pKRHKz8n35J6OmFWz4TKGNth
n87D28kmdwQYH5NLsDePHbfdw3AyLrPvQLlQw/hRPz/9Txf7yi9Djg9HtJ88ES6+
ZbfE1ZHxLYLSDt25tSL8A2pMuGMD3P81nYWO+gJ0vYV2WcRpXHRkjmliGqiCg4eB
mC4//tm0J4r9Ll8b/pp6xyOMI7jppVUCAwEAAQ==
-----END PUBLIC KEY-----
```

which decodes as:

```
algorithm: AlgorithmIdentifier{id-composite-key}

subjectPublicKey: CompositePublicKey {
  SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: ecPublicKey
      parameters: prime256v1
      }
    subjectPublicKey: <ec key octet string>
    },
    SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: rsaEncryption
      parameters: NULL
      }
    subjectPublicKey: <rsa key octet string>
    }
  }
```

The corresponding explicit private key is as follows. Note that the PQ key comes from OpenQuantumSafe-openssl and is in the {privatekey || publickey} concatenated format. This may cause interoperability issues with some clients, and also makes the private keys appear larger than they would be if generated by a non-openssl client.

```
-----BEGIN PRIVATE KEY-----
MIIFHgIBADAMBgpghkgBhvprUAQBBIIFCTCCBQUwQQIBADATBgcqhkjOPQIBBggq
hkjOPQMBBwQnMCUCAQEEICN0ihCcgg5n8ALtk9tkQZqg/WLEm5NefMi/kdN06Z9u
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDbFUCtX3C0Dmy+
GWLXM1mzmxAgBUrJ4NDi6UUz6/VRld0rEv1aDEWalxfrLM8+iNlHBU2jBfQsDUW1
i7iAXM87oWx0o5+uTrqDfjptsLD6PqJMOOkU3pAeGnfUEL2EVcz5SC1EPDb2eFtA
Si74l011iIII7af2NaSkRys/J9+SejphVs+EyhjbYZ/Ow9vJJncEGB+TS7A3jx23
3cNwMi6z70C5UMP4UT8//U8X+8ovQ44PR7SfPBEuvmW3xNWR8S2C0g7dubUi/ANq
TLhjA9z/NZ2FjvoCdL2FdlnEaVx0ZI5pYhqogoOHgZguP/7ZtCeK/S5fG/6aescj
jCO46aVVAgMBAAECggEAFtT6LpdZuYofTxh6Mo9Jc+xfG9cxWiSx4FQLQEQBBwWl
TQ3nlXDd+CRy+7Fpz8yXSE2HL8w5DDY945OyIL6LYl2KXgWHaLUPvxByqmfVqd7J
L0RnFiOzxU9g2Zr9BUOj3v7kqM3VtI4KhIK2rnWmPu+BDckmzgP9Kpm4KhbPuAYP
iqUZSkxpSUsd5ALLsk9b0xjR7UEYkEpV2/vORwieEhOmPLzuXh+Px0yavkazT/vU
+h/rDSoLQn7v4fVsQgNdOaaOG/gHemGuuiLPJJlX5ZZ6mmsIaEjz+MNk0aJDH2po
KbAr4B709dTsnYgv7YtkEfSyOeMEdhMiswI1c9FpwQKBgQD6kdHmHCoeWNNvlqxU
v57e7ZDAXDA6WcfrypcsF0l72rI3J8oOPmFaNaCmwIH/Icz+Zy7fr2IYxVjyDjCa
zi8qTnj2ZNds71hUYOcq60u0TcSVrtocA4HW7NoWJqK5thNlNaa1M358cYBopGoN
ocS9yf10q2MBZtpF0fc5PbFf+QKBgQDf1L4cezoebbNTaN4KoapycHXxKozP2GwI
r15YRYjt0ZpHstdUPABQuwlL9CuL+5Q17VRiM81cUVNfFsBzKIXYb/PBC5UD+DmR
qGlT6v6uUWY6jifUgEjfyPxO0oJ3M6cChHR/TvpkT5SyaEwHpIH7IeXbMFcS5m4G
mSNBECO/PQKBgCD0CoHT1Go3Tl9PloxywwcYgT/7H9CcvCEzfJws19o1EdkVH4qu
A4mkoeMsUCxompgeo9iBLUqKsb7rxNKnKSbMOTZWXsqR07ENKXnIhiVJUQBKhZ7H
i0zjy268WAxKeNSHsMwF4K2nE7cvYE84pjI7nVy5qYSmrTAfg/8AMRKpAoGBAN/G
wN6WsE9Vm5BLapo0cMUC/FdFFAyEMdYpBei4dCJXiKgf+7miVypfI/dEwPitZ8rW
YKPhaHHgeLq7c2JuZAo0Ov2IR831MBEYz1zvtvmuNcda8iU4sCLTvLRNL9Re1pzk
sdfJrPn2uhH3xfNqG+1oQXZ3CMbDi8Ka/a0Bpst9AoGBAPR4p6WN0aoZlosyT6NI
4mqzNvLE4KBasmfoMmTJih7qCP3X4pqdgiI0SjsQQG/+utHLoJARwzhWHOZf1JKk
D8lSJH02cp/Znrjn5wPpfYKLphJBiKSPwyIjuFwcR1ck84ONeYq421NDqf7lXbvx
oMqjTPagXUpzHvwluDjtSi8+
-----END PRIVATE KEY-----
```

   which decodes as:

```
algorithm: AlgorithmIdentifier{id-composite-key}

SEQUENCE {
  OneAsymmetricKey {
      version: 0,
      privateKeyAlgorithm: PrivateKeyAlgorithmIdentifier{
        algorithm: ecPublicKey
        parameters: prime256v1
      }
      privateKey: <ec key octet string>
    },
  OneAsymmetricKey {
      version: 0,
      privateKeyAlgorithm: PrivateKeyAlgorithmIdentifier{
        algorithm: rseEncryption
        parameters: NULL
      }
      privateKey: <rsa key octet string>
    }
  }
```

B.2. Explicit Composite Public Key Examples

B.2.1. pk-example-ECandRSA

Assume that the following is a defined explicit pair:

```
id-pk-example-ECandRSA OBJECT IDENTIFIER ::= { 1 2 3 4 }

pk-example-ECandRSA PUBLIC-KEY ::= pk-explicitComposite{
    id-pk-example-ECandRSA,
    ecPublicKey,
    pk-ec,
    rsaEncryption,
    pk-rsa,
}
```

Then the same key as above could be encoded as an explicit composite public key as:

```
-----BEGIN PUBLIC KEY-----
MIIBkTAFBgMqAwQDggGGADCCAYEwWTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAATE
Y+Gue5Ib98bLDUU36XmHingAZFAJLQsFecE752OFwjqqcXZO+tARohgYalBPnAdL
7vLmIkn50TmqPFy/yWhEMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
2xVArV9wtA5svhli1zNZs5sQIAVKyeDQ4ulFM+v1UZXdKxL9WgxFmpcX6yzPPojZ
RwVNowX0LA1FtYu4gFzPO6FsdKOfrk66g346bbCw+j6iTDjpFN6QHhp31BC9hFXM
+UgtRDw29nhbQEou+JdNdYiCCO2n9jWkpEcrPyffkno6YVbPhMoY22GfzsPbySZ3
BBgfk0uwN48dt93DcDIus+9AuVDD+FE/P/1PF/vKL0OOD0e0nzwRLr5lt8TVkfEt
gtIO3bm1IvwDaky4YwPc/zWdhY76AnS9hXZZxGlcdGSOaWIaqIKDh4GYLj/+2bQn
iv0uXxv+mnrHI4wjuOmlVQIDAQAB
-----END PUBLIC KEY-----
```

which decodes as:

```
algorithm: AlgorithmIdentifier{id-pk-example-ECandRSA}

subjectPublicKey: CompositePublicKey {
  SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: ecPublicKey
      parameters: prime256v1
      }
    subjectPublicKey: <ec key octet string>
    },
    SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: rsaEncryption
      parameters: NULL
      }
    subjectPublicKey: <rsa key octet string>
    }
  }
```

The corresponding explicit private key is:

```
-----BEGIN PRIVATE KEY-----
MIIFFwIBADAFBgMqAwQEggUJMIIFBTBBAgEAMBMGByqGSM49AgEGCCqGSM49AwEH
BCcwJQIBAQQgI3SKEJyCDmfwAu2T22RBmqD9YsSbk158yL+R03Tpn24wggS+AgEA
MA0GCSqGSIb3DQEBAQUABIIEqDCCBKQCAQACggEBANsVQK1fcLQObL4ZYtczWbOb
ECAFSsng0OLpRTPr9VGV3SsS/VoMRZqXF+sszz6I2UcFTaMF9CwNRbWLuIBczzuh
bHSjn65OuoN+Om2wsPo+okw46RTekB4ad9QQvYRVzPlILUQ8NvZ4W0BKLviXTXWI
ggjtp/Y1pKRHKz8n35J6OmFWz4TKGNthn87D28kmdwQYH5NLsDePHbfdw3AyLrPv
QLlQw/hRPz/9Txf7yi9Djg9HtJ88ES6+ZbfE1ZHxLYLSDt25tSL8A2pMuGMD3P81
nYWO+gJ0vYV2WcRpXHRkjmliGqiCg4eBmC4//tm0J4r9Ll8b/pp6xyOMI7jppVUC
AwEAAQKCAQAW1Poul1m5ih9PGHoyj0lz7F8b1zFaJLHgVAtARAEHBaVNDeeVcN34
JHL7sWnPzJdITYcvzDkMNj3jk7IgvotiXYpeBYdotQ+/EHKqZ9Wp3skvRGcWI7PF
T2DZmv0FQ6Pe/uSozdW0jgqEgraudaY+74ENySbOA/0qmbgqFs+4Bg+KpRlKTGlJ
Sx3kAsuyT1vTGNHtQRiQSlXb+85HCJ4SE6Y8vO5eH4/HTJq+RrNP+9T6H+sNKgtC
fu/h9WxCA105po4b+Ad6Ya66Is8kmVfllnqaawhoSPP4w2TRokMfamgpsCvgHvT1
1OydiC/ti2QR9LI54wR2EyKzAjVz0WnBAoGBAPqR0eYcKh5Y02+WrFS/nt7tkMBc
MDpZx+vKlywXSXvasjcnyg4+YVo1oKbAgf8hzP5nLt+vYhjFWPIOMJrOLypOePZk
12zvWFRg5yrrS7RNxJWu2hwDgdbs2hYmorm2E2U1prUzfnxxgGikag2hxL3J/XSr
YwFm2kXR9zk9sV/5AoGBAN/Uvhx7Oh5ts1No3gqhqnJwdfEqjM/YbAivXlhFiO3R
mkey11Q8AFC7CUv0K4v7lDXtVGIzzVxRU18WwHMohdhv88ELlQP4OZGoaVPq/q5R
ZjqOJ9SASN/I/E7SgnczpwKEdH9O+mRPlLJoTAekgfsh5dswVxLmbgaZI0EQI789
AoGAIPQKgdPUajdOX0+WjHLDBxiBP/sf0Jy8ITN8nCzX2jUR2RUfiq4DiaSh4yxQ
LGiamB6j2IEtSoqxvuvE0qcpJsw5NlZeypHTsQ0peciGJUlRAEqFnseLTOPLbrxY
DEp41IewzAXgracTty9gTzimMjudXLmphKatMB+D/wAxEqkCgYEA38bA3pawT1Wb
kEtqmjRwxQL8V0UUDIQx1ikF6Lh0IleIqB/7uaJXKl8j90TA+K1nytZgo+FoceB4
urtzYm5kCjQ6/YhHzfUwERjPXO+2+a41x1ryJTiwItO8tE0v1F7WnOSx18ms+fa6
EffF82ob7WhBdncIxsOLwpr9rQGmy30CgYEA9HinpY3RqhmWizJPo0jiarM28sTg
oFqyZ+gyZMmKHuoI/dfimp2CIjRKOxBAb/660cugkBHDOFYc5l/UkqQPyVIkfTZy
n9meuOfnA+l9goumEkGIpI/DIiO4XBxHVyTzg415irjbU0Op/uVdu/GgyqNM9qBd
SnMe/CW4OO1KLz4=
-----END PRIVATE KEY-----
```

which decodes as:

```
algorithm: AlgorithmIdentifier{id-pk-example-ECandRSA}

SEQUENCE {
  OneAsymmetricKey {
      version: 0,
      privateKeyAlgorithm: PrivateKeyAlgorithmIdentifier{
        algorithm: ecPublicKey
        parameters: prime256v1
      }
      privateKey: <ec key octet string>
    },
  OneAsymmetricKey {
      version: 0,
      privateKeyAlgorithm: PrivateKeyAlgorithmIdentifier{
        algorithm: rseEncryption
        parameters: NULL
      }
      privateKey: <rsa key octet string>
    }
  }
```

B.2.2. id-Dilithium3-ECDSA-P256

This example uses the following OID as defined in Open Quantum Safe, which correspond to NIST Round3 candidates:

https://github.com/open-quantum-safe/oqs-provider/blob/main/ALGORITHMS.md

```
id-dilithium3_aes 1.3.6.1.4.1.2.267.11.6.5
```

A Dilithium3-ECDSA-P256 public key:

```
-----BEGIN PUBLIC KEY-----
MIIIKjAMBgpghkgBhvprUAUBA4IIGAAwgggTMIIHtDANBgsrBgEEAQKCCwsGBQOCB6EA2wEI
X+0DRUsEixdVXp+ZVcigmaDEEDNCSblD9t5nERtIufPxKONt/IRXok6v5jKWbZWIFB6ZT3Ev
crm2Qybxvrp5GwB4FaUe+0IwuMlrt7Nr/57WTqgxeeBMWbHnudqX2QzP2nnlAWip3YQ1hWj2
Z1qnRzc2C8S+eR56qcAPd+xN8ehs/n1WzNQoBigXXbJsh0zpVZuML+p15GEH7JvEtV+4flx6
CkEahgXLXZFJ2i7gDa+tWfXPO1oa1cb9uhLEkZjEsy9YivBOmmEhwq9pEterizqrqffP1EIc
BB3Q5mqhJTh42+i1oRVDrcN0dL6ZIc26NI0ebbBKUuHZLlMPppsF8x3jg88uZjshvwaQy3cl
pcwRf48/nbMiSaaWvl0/dXNgrZqYPrvUPGIFr+j+YMLVod2+74tum7AOc/k9/SsUcoQf1BkF
5+ViK1n+CwdvbBS4VPeoTroO6YiPTB6sT+id3YIgI03yQmOQJTsRr63Sq7a+f+IBwVPuo1ZM
MGngXtkquTEV/Ufs0sP6D5KmOs5lhGVe8V61h+N2r9MLuKIlV7CaoBGGTQ1AXj0vliWDltZn
FRJ4QYQFsyln7DVORfE+luuq1YmutyoC3G/JgnSjtoGOp6NQKrQGWgCV0gNIYa/Khg9cPcII
82LCVNRhRdI+hPkIHNcLNlJyBQ9A3IHHERtuJDQMoJN6aq5yx1rhbOUdGhWZEZHiX9FaLcUR
bUPVdCZrmLuct/5rbnfMfc0bmmHR5cRmENMEIwI/kLljrCoDmcMEGfSBZEEqX2I5+xVAyUfm
baYXlzkTjSX4WXXx70pkCY7QNymW6im+/neA78w3TMV7P+0dqehuA6ohC+2puIkyZ8zOm0vf
7vFXlqh2eV38TBVT0umIaw5mvv07+9S0OOzdDrR3iix1MWA4iT00h3wi1y45KHrbb6blZHEC
f1crFR03SpjFqiSkjGezxWN84lPMTZKQ/dflmxWhN4yYVJpVadSwJoY9atRVI0QI1cdvO2Af
IwRtrauRrYdEVkAkr5H65nxBkU2clHsIdQDzTB+7npX0KUsokKFIe+Yb/Ofko7Zz1jHBaaPi
6Ul1w8oRWzqREQPWeKRd1SEziC21DZ4MXIuzZvpH7/W+LqXzC9DvaJOCHY7fve9cMWDW9AuH
Mtk5/NnekpUKit4acfIFn/YcUa3gN3uUqoC2x2dZ/GrehDM5Ff8I/b5fqxBB42tJbUWIHwxI
2iXGn/dpKxV/IRtvo8VRlRgwaKILIn9MjJxaugWzsVvxDuxDPGB9WgnelYOgVL8lt6l8RrSf
vdIDfwm0rzd0xQMGLwlynCtIOY77XRyOl1SR3crHjNx4H+R7nqF6jMGBS59nvlGF/CC7OX84
TaUXPk8hrR4RB7elhBR9azgadnqDfAbvoHmjv46xRniCVJywrgkeU73FzU1WRLT8jrECDLln
pXMyMB6L4I5rrhQ6R3Hc095SnMcSWTmFSOwFs2hsrkzEfJhWZXRhCeKWpe0zFhHRz4MUY5Rf
TabYOpRWhMK6CgA2l5tFc7r+SKxK1v3i8mZ3sB2TptIlfSVU+iQT8RLEdJjWC9BBqW76a03u
iUr6aKsw2z+AN/ZnuLyZ3YvWLHNdCRmtZlASMZfoa5K+FfkXUog4m0IzxHzy2dMOpDXsi489
+8D3L51Le/K9FR4xjLo8TMAjnoS2KMrKfmiseIqIMJfcECcKyM8UJrECkS+GC3Hu9g22bsyo
LwYu7gKbElItj4dlj7RRv7DXYPF/fJiUkcjK8WTQ9geHKrcooe8DMMMNb14Z5G27P6ecZaaH
M1ixnkIF7EMXF4EVlXo4dgSF0Ozc+LF57pYVc9x//5kprQ35Va8Cs1YJTooS2bIuZEzQx9Gx
LG4yQvbp2q3dQxQKqovdOp/c3yZfFffBQsOPCUKsnmpEJsKhUrNEfzuylZTk5/8gnH4//6wG
kq+2VLi2YNBj/YCBFRKTQLqUkiZKRjeyWakBo6aVS+rJS/gQD08DMISiihNK6Xi0bxoHV0le
05q3O+QuZEjPjZp1w33nwv+ck3uqPsH8HQbvEQnkxFidEK/9vXQyB2EZkWfTYuNvWPRsBab2
Om0nUh6Kg+xHZZrX7hfN7GMvAx7jfBSYoHSCuRhZuSMYwXYWadBiV4KsCNAOm5PqE4p6DQmY
m8/3TzkR7m4fEuF/E5qHZ2vpkWI4MTGx3uYXf3dLHd+JiEorXL+HzR/dPhaHTDL/XwenhrGZ
/YNAXka/6OpzN/+oP7zmEjLcs9RKqtRlipmjlusriUfxJ5wxi+xGS+xyrjXeXlqe3EV0JVdV
0iqR94g9PX69zuBP/rPNxdQf3od86mreh0ts5ul83KmUK0xpTOQqLAZUuckeC6pmA27nZOdj
hySgzTu32gAwXyifRQc0wE/EmZWRneLgh45QecDE+glcGY8rn0+P5dMyNhlNwNwbgpc8NSQa
dM9Vq8+WtyBRwSjG/TkjEMY6Ex+hIASoM2D4Hs5MJ0AHNswWTATBgcqhkjOPQIBBggqhkjOP
BwNCAATLQgUt46GGb96T815H65I05ELw9siX91ZlN+vVNW9RLQm519BPfX+1Vk+HAviXMVk0
Srzd4w0i+5dCNM9rX
-----END PUBLIC KEY-----
```

which decodes as:

```
algorithm: AlgorithmIdentifier{id-Dilithium3-ECDSA-P256}

subjectPublicKey: CompositePublicKey {
  SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: id-dilithium3_aes
      }
    subjectPublicKey: <dilithium key octet string>
    },
    SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: ecPublicKey
      parameters: prime256v1
      }
    subjectPublicKey: <ec key octet string>
    }
  }
```

The corresponding explicit private key is as follows. Note that the PQ key comes from OpenQuantumSafe-openssl and is in the {privatekey || publickey} concatenated format. This may cause interoperability issues with some clients, and also makes the private keys appear larger than they would be if generated by a non-openssl client.

```
-----BEGIN PRIVATE KEY-----
MIIXugIBADAMBgpghkgBhvprUAUBBIIXpTCCF6EwghdaAgEAMA0GCysGAQQBAoILCwYFBIIX
CF0DbAQg06Bf7QNFSwSLF1Ven5lVyKCZoMQQM0JJuUP23mWccgkCYK6iOhDmkqybIASjgzCU
FDOEPKot4rEOPa3J+Wnp1odULvCAUvzIbi9DKIk2xBJVvJI6oS+WBq2JwAE0NgVHZ0BBhUYk
CMyNxIxF4QRE4YXN2UjKBhBcAYTBHFyBXYVJ0Eod0CGIzGGCERBFhdXdRERMmeIB1FgM3EQc
cgU4MlACVldTh0NmRANSUkAEFIVBdzRgEVFHIgFocRZYUxJERUE4gCg4KANkaHESJwgCYFFD
4gHhmAmYDERd2NWc0MIhogBSBUBFkZBhjBTIkVXdVNEWDSDVVUTVGcodzSIaDOEEIMVFQUgV
IQcCBAEWVXYmUFaGICcyInZjMgQIZIUiYmghEwdGeEgShTWEQ0KHAnc4ATAkNjBYVEFWUUSD
BCENTeCh1VXY2RCY2JEJYdDQXMUeCIRAhEFEDNiJxIWQhMjAAAIU0UCVCEBIxUBZRhDUnMkC
gYUjESZ3FydCMVBFZ0YxA3IwFRcxQ0d4UQWGQBAYOGQzdSQyV0EldWV1A3ZmNTYmaCBxIjUo
IQYEoNnMAZANHJBQFZQIxNxiEVQIYCDElGEAHYlUlFFWDVIMDRgZUJzgVAwdnQUgicAFwRUM
VDZkEGRYISMXJRYSaIF2UAADcSSHB2RHAEdzCFF4ghRwdyUlUBAzhGd3BHNnFGAwMHZxR4ZQ
YAVUHcoQQMFcGEohGVQBzNVZTQBRiJVdSgmZXF2dTQxFnRkIWBDFFeANEMnEmJDFVVxCGUmV
JCRHIDUlESE4iEFIBAZoUhFmcEgHIBA1JlAUiEAhRXIlFUVDIkYXeGI1IxcoJCd3dWB1FHA4
BRgN0JhByc0eFUVQlBRYIRhIoQBdTNnE0YxVYCDZjY1gAeDFlc3GDVgI2FoIYMSOBZQhIgBe
iBVgMDYoZ3dzdDFxFjOEUSREEldTV2MQNEhShTYkNYIFMgNwJyBGIyUVJnVoEzhEEWF1KHFV
FUEF3VhdmIwMVABh3dzMDUIBTGIQDFRN3doNhhjdldGgjhUZXgUQRg2IGZVZkBBQABGgEUAh
hTZliFAxFwcnQVUVZ2IQJCGCISYxQncFAnJ1cxN4gDZ1ZlIRVjM4UIRIRTAnIRd3ZCgjdlaI
mBlVQNTQocIKACBhTI0NhY4NVElBgASB4WDVYQziGAhRyR4QDdChhZ2B1ZBFFciUxIQYFJlA
Q3AQhoQAcGByOHNIYGiEI0hlERJxghhkIyYRMHGIFmYoZieDeCBXMhd0IENTQlGFFygQdyBm
CITEQFXBBUCUxNVVigGgoEVgDByhwQVBHcicCJoWEKAM1NWcFJCYyJBRzcjNARHEUYGgoUkV
F4MoaEZ0gUODMQeCQwAySCIiJmhgBmBxJVFgJVZyJ4OEB2QQdGdRgSJUB3Z2BBMHECQ0YXCE
RWBAFBWUiAyM4hwIwUkBgFwc0I3QlhGYxECR0QQVUhRIUcjEihgYxiAQyBySBciB2SHQoJRV
cyUAc1N2hRhQMGBVQwJFeAcVNTA1gEggMxSEUGd4Y3FWSFcnRRAFGEUldChUcwWEFnEigVZn
mVCE0CAMkeAFSEWWHUShEBXYmUAgVQkhSZCaCU3BCdFMlZnAWUncwJShThxYyBTgzcYdoVkJ
EwV0BmdxAkZwAxMGCBgBZYKDdzElGBEghBUwIzBDgFA4OGBXICVXFWBSd1glEygSVDJVdYBk
mAlgkNUJQFoUXQoaHUmR3CEJ0KDQWRUiChHMXh1ZgATMBJGZGFhIiFEcyABJmZkSFY0UldId
VWMmhUEocRhigzJAZlOCEhAyB3EHZGUEFDU1GIeHYjNgNII2dwN2JYc3NFWIcycDJjdISCOE
BCGElgoUYMECBAIMEcGdGWFWFc2h0eGERM2Byc2Q2UwQXdWImZFKCJhRVaFOZ0+KDXUoj30j
YL4L7KEAuevyOLifPNDnBY/rrg6GB81xNpsdKdjg+osRG0OsBNsmnxkg3nALFodEYQ7Z9Qyx
5rreYJ0oy77dXuc4DoHaOxtPuAMOn6aKZqBtUktyInranRPjubOSe6isnInZk7IMom8egOvd
TguF+WaiaFU0WCZIghuUQxYqGJ34QmRLSwLT4f+uHoIV059jAzuG2E4c+2bGyebTCKkzkahM
itzln03fRygDMTJREy2uvzxb/pAZ4Tewp9tVlP90Lf33cyqvwJBlj+yv7HeRneg6g/6GOs/4
4k2Tbk7iZdn7fGnjqf3Sdiz7pNECCHvAF/TPnxCYA41wUD+gNUnaiz8Npk9N02V0FXi7NZKJ
lVtIuToWlVHFOx9gswYiCmUS56mqTiLs/KR8c4zF14NnvQiakpl60tKntDkcBUZgoruDJLnL
yft7TQlnRb4NG0hjv7s1uY/EnT3mvgGHIJu3Tb2sqlHLkVLsK19rsMqbF689svN6WCZ38jNs
FPEzcCE+fb91n0mWx+f6Ab7OSm/CPS1WmoMHxi3ad2QT93f3EJqfNjOZiiTDKPre/ybYf8EB
Yv8YA+HJn7zduecMIu0tEVkEnUqTm1Jo7UJPXO//R3TSBgcnX9AwBgRagP6/gArBVh9j9Bjd
fmS1bUY/Bz4Yhoal1X1rpTfSzsw/ZXcEpM/cYRUEPaFNsNcFUjUi4t8e68fp/CciGXkLEl5L
72or4gCSzrzOaPwnSB/HVSflRtQPqD6i9QbR71QhgvtF+KeZD2U4iLaJf1fDUNqXa+IXNy64
cMuQN28uB0ewxUn68UdRMUIjmq1kNoHr4QH6J6IEhC6ejjbNkDPvoRrIuoIsJ1rTPoVPEpzT
cNlqWjRHHPtT7qH08xIgP0Hsa+8qa79vBkmqIUcRFtNdhGfv06pFwvJa0B9dTSxfXIO8iO3L
Daj8W44gwx6m5MXaMLXnjcwH4i7vf8k47RpifCr8T2V8D9iCKNdFfd6FdHEZfkCjbe7/aVU8
2AUrGYCpgC5MBkSV9IoxT05KTC0ut82Ni0BRDi+LsdI3syNoARoE0BqQTuF5XEFJVe6MXzmZ
XfQK8aW6zZnHwW9TtlF23YR9ozDN0fFkwsoMd8qeFsgzUOo1TYi8w1P4FnmLp0NdoRDf/gX+
Fs0CWSYZAYubIR5cgDPN0mwlQJDW46sr2ATRabjJ0A/N9ZhxdWEV1Qfb91z3JixpUGjQ1cPh
ZnJKCMarCmx9rNjoGo/WJ1iOilahBihCH83k+hUWFwNNrTZSm+TnzlObpu9WgE5g0e/QzQrs
o3s25BQqJAGiRKBLd0UXz5AUbSB9fFaO21aiX2WWISjdA1R4SuQNNQZcVgGuKpQBAcPV7xcF
eyCOw3/AA+WInx8mHjtDcOIowbEG1HQzV4CTsAelNp7I8CtXsXL539wiUpO8A2Fbh2wqTbe1
```

DSM7VPezIBE9gFPIDGJul1YyyzGON0ukKAHWmAiUuoBtlrryFFEO1Cnh01aoP41vMWMiqM9X
/oeAYdAyY21G3oyKV4JXrNufFNyhivCRGaptCKDJ15odOe4ZXJ6BIaU3UBz+cm85OOAwHq1F
XwEuLU95G65QHzukC42ajwAhVLjcRudqNZ0Nb4vi63NiP+ZVrS52vZJGSPa6B7Vss+PYsZKp
j54Y5Z1eU+Q35lnLLSKMOkLTkCP3oFxX8mW5hnZO3ysCkytBC3CRe+7dxTMP0XWDT5LraSph
7ocMAKh8mSPYhqAAaAgekjmvPDMbIezv5slR4M1/br5cc4z/01UwKV2qkT8DQxM9vkHO5XLo
DRJXW1hUnpjhOrOzMsiTJ6aQ/+z4dBjJxR9a9mYfkju/SEk3V8TX7VmKQk7WpKTt6b2xGiQc
5BZBouYeNZ1D9Bgv6CyME0GIjf9nIXyq3mNs4ALGgheo+RIcPXrkhVZgdZexEtl0N6s0ynZg
o/baP1pV730vAqIgbwSerx6773sqSorU+L1HL1XQrm4EjgzirKn9/IMm+gz9zYRZy1RdnnKv
+mV85fOOlUJZZhdyOKEXp+OJnez70Y7iHnmuqvDlX8aQXzPm+ehXohw4yiLVhcENTQyGRjR0
3SkW0OyuVMysYMo6LBAejaoKdCbcI6VmRZOPRWXbhch2Pv29SErLsbJrSSE0JAdRIZZKW6Iq
oGQ9oB72br5mSFeg5tYMdDSgqyOVqJmFNB6PzEvKbke8AwAiaZSX7tteP8LVxkIJPHIaBAVV
olG2Oe/eDXbyz0ZwbomnHe7wzyGc3W3LmpbnP5wF2QnIRRrLnBQgwvMpV/JuGkdcWc4awiSw
cvAhQKCUWvLYgGLNaa8fh61OVOGCKymMt00w7/RGufPKaunCzlei9JF0IKCZBS0LzOeVMhP9
jjopJSP6Mg1RPCTHD1caQ6NT6HVrv08SmHCRG+G0szAEpC72sXjeYkra+1bwj6RCw49uzd4g
Ruv9/iN5tx0CW6em7bH1K30UFMDEB/zTsEBSj7QkdZ7TLItDIUY/ztMu+La7zHfCFIz7OpN7
xLDax+pAC4mBunet/zOyaIc1XJozwTz+HmwAorOlEqPklXNJ+78VKtDih92px9K0ALX70/ak
kLoxqf2NfLEd3YOJhUZE3jt3KYTpqohzJWAG9e+2FNXMqSrTOY8YZq4HThTxyWjvNUX6iQYc
N17XenbHp1Cfdk4Bc8mTheUocqUOtrRFWsO+XhPJXSv6XXwPgbyD0Jr4xMQ7ni/vfEZuR6qb
+Hi54aEsF+2Y4WfafOFQvRNcemC8J2rMJenGbCzfysU1wiz86qizoGocbd3OS9GqPy29dupf
t0dWnFmm7SHoqjtpbyjr31Q9zSt271gxepriull6rVegURHtZtINi6LHL6yYCd3PQcFdm7tU
D3bAjrvOIf23rdQpahyIAE+xrtamDF/0qjirz8XBMkNxKgXbA5hgyhBmvDU/YjGxIhyOZj7t
XnyDiUwdxO/i0onKxh56aI9sPUptY8AGOArV/1DZ5gJqjcF9BMs1CTZTGQiReyL/stfgkDcU
a3+K8beH93CNe2fYYOPIVc02fY9kD85HW1ljtsm6caRAHZl0IxXWpeGbmutqhuWyWe6uVpP2
LPkJFRnpUFcVkJbJs6nUspUdLE4sltzEjykqjPoBKct7rGRSCK6A7jBTrqArh7yYPMcm34bM
GjvT9pzm8sws2wEINOgX+0DRUsEixdVXp+ZVcigmaDEEDNCSblD9t5nERtIufPxKONt/IRXo
5jKWbZWIFB6ZT3EvIpvcrm2Qybxvrp5GwB4FaUe+0IwuMlrt7Nr/57WTqgxeeBMWbHnudqX2
2nnlAWip3YQ1hWj288bZ1qnRzc2C8S+eR56qcAPd+xN8ehs/n1WzNQoBigXXbJsh0zpVZuML
5GEH7JvEtV+4flx6iPSCkEahgXLXZFJ2i7gDa+tWfXPO1oa1cb9uhLEkZjEsy9YivBOmmEhw
EterizqrqffP1EIco+LBB3Q5mqhJTh42+i1oRVDrcN0dL6ZIc26NI0ebbBKUuHZLlMPppsF8
g88uZjshvwaQy3clEhPpcwRf48/nbMiSaaWvl0/dXNgrZqYPrvUPGIFr+j+YMLVod2+74tum
c/k9/SsUcoQf1BkFYV25+ViK1n+CwdvbBS4VPeoTroO6YiPTB6sT+id3YIgI03yQmOQJTsRr
q7a+f+IBwVPuo1ZMI/4MGngXtkquTEV/Ufs0sP6D5KmOs5lhGVe8V61h+N2r9MLuKIlV7Cao
TQ1AXj0vliWDltZnVQ9FRJ4QYQFsyln7DVORfE+luuq1YmutyoC3G/JgnSjtoGOp6NQKrQGW
0gNIYa/Khg9cPcIIMRn82LCVNRhRdI+hPkIHNcLNlJyBQ9A3IHHERtuJDQMoJN6aq5yx1rhb
GhWZEZHiX9FaLcURoChbUPVdCZrmLuct/5rbnfMfc0bmmHR5cRmENMEIwI/kLljrCoDmcMEG
ZEEqX2I5+xVAyUfm3j5baYXlzkTjSX4WXXx70pkCY7QNymW6im+/neA78w3TMV7P+0dqehuA
C+2puIkyZ8zOm0vfbte7vFXlqh2eV38TBVT0umIaw5mvv07+9S0OOzdDrR3iix1MWA4iT00h
1y45KHrbb6blZHECBvOf1crFR03SpjFqiSkjGezxWN84lPMTZKQ/dflmxWhN4yYVJpVadSwJ
atRVI0QI1cdvO2AfDdPIwRtrauRrYdEVkAkr5H65nxBkU2clHsIdQDzTB+7npX0KUsokKFIe
/Ofko7Zz1jHBaaPig0u6Ul1w8oRWzqREQPWeKRd1SEziC21DZ4MXIuzZvpH7/W+LqXzC9Dva
HY7fve9cMWDW9AuHvAPMtk5/NnekpUKit4acfIFn/YcUa3gN3uUqoC2x2dZ/GrehDM5Ff8I/
qxBB42tJbUWIHwxINT+2iXGn/dpKxV/IRtvo8VRlRgwaKILIn9MjJxaugWzsVvxDuxDPGB9W
lYOgVL8lt6l8RrSf7cEvdIDfwm0rzd0xQMGLwlynCtIOY77XRyOl1SR3crHjNx4H+R7nqF6j
S59nvlGF/CC7OX84ZxgTaUXPk8hrR4RB7elhBR9azgadnqDfAbvoHmjv46xRniCVJywrgkeU
zU1WRLT8jrECDLlnLVvpXMyMB6L4I5rrhQ6R3Hc095SnMcSWTmFSOwFs2hsrkzEfJhWZXRhC
pe0zFhHRz4MUY5Rf0wwTabYOpRWhMK6CgA2l5tFc7r+SKxK1v3i8mZ3sB2TptIlfSVU+iQT8
dJjWC9BBqW76a03u7YqiUr6aKsw2z+AN/ZnuLyZ3YvWLHNdCRmtZlASMZfoa5K+FfkXUog4m
xHzy2dMOpDXsi489oiX+8D3L51Le/K9FR4xjLo8TMAjnoS2KMrKfmiseIqIMJfcECcKyM8UJ
kS+GC3Hu9g22bsyoJHELwYu7gKbElItj4dlj7RRv7DXYPF/fJiUkcjK8WTQ9geHKrcooe8DM

```
b14Z5G27P6ecZaaHL6FM1ixnkIF7EMXF4EVlXo4dgSF0Ozc+LF57pYVc9x//5kprQ35Va8Cs
TooS2bIuZEzQx9GxXigLG4yQvbp2q3dQxQKqovdOp/c3yZfFffBQsOPCUKsnmpEJsKhUrNEf
lZTk5/8gnH4//6wGD4mkq+2VLi2YNBj/YCBFRKTQLqUkiZKRjeyWakBo6aVS+rJS/gQD08DM
ihNK6Xi0bxoHV0le7Ag05q3O+QuZEjPjZp1w33nwv+ck3uqPsH8HQbvEQnkxFidEK/9vXQyB
kWfTYuNvWPRsBab2f+iOm0nUh6Kg+xHZZrX7hfN7GMvAx7jfBSYoHSCuRhZuSMYwXYWadBiV
CNAOm5PqE4p6DQmYxutm8/3TzkR7m4fEuF/E5qHZ2vpkWI4MTGx3uYXf3dLHd+JiEorXL+Hz
PhaHTDL/XwenhrGZSZc/YNAXka/6OpzN/+oP7zmEjLcs9RKqtRlipmjlusriUfxJ5wxi+xGS
rjXeXlqe3EV0JVdVSJx0iqR94g9PX69zuBP/rPNxdQf3od86mreh0ts5ul83KmUK0xpTOQqL
uckeC6pmA27nZOdjoMUhySgzTu32gAwXyifRQc0wE/EmZWRneLgh45QecDE+glcGY8rn0+P5
NhlNwNwbgpc8NSQaL5+dM9Vq8+WtyBRwSjG/TkjEMY6Ex+hIASoM2D4Hs5MJ0AHNswQQIBAD
gcqhkjOPQIBBggqhkjOPQMBBwQnMCUCAQEEIBqglVGEbVW2JdupT30vKPECOx29/9JjP8kbw
9wve
-----END PRIVATE KEY-----
```

B.2.3. id-Dilithium3-RSA

This example uses the following OID as defined in Open Quantum Safe, which correspond to NIST Round3 candidates:

https://github.com/open-quantum-safe/oqs-provider/blob/main/ALGORITHMS.md

id-dilithium3_aes 1.3.6.1.4.1.2.267.11.6.5

A Dilithium3-RSA public key:

```
-----BEGIN PUBLIC KEY-----
MIII9TAMBgpghkgBhvprUAUCA4II4wAwggjeMIIHtDANBgsrBgEEAQKCCwsGBQOCB6EAD7Kv
fJ66NzesNkOiHBXWcF5FYs9mBugtOHAUrl86Ns8l24jV6Ut+TjYd8TUbUClNoWhGe/v2W/gZ
QGlUahMxLY68nqH2BXO5bjMHbE4pGGGuNejGJnJoe+1/kd9+Mym6LE2YpAZCKRtgka9wXR3i
C+qp9OHi2tt9cpur11mk6NPWjdVMqXxOBYgeWESR5k8fYS6ttRNPEC2JQ5ZtNMV1srdov+mj
TmGIXWUwXWEI/5LQAthtaSywFPHj96+kLlm79lbYeaVfSN/URojQZLAq2gx0DFK8m5ZA9GaR
L3eavfzw2Kd6pOdFNemTquM/i7uGFc4/Tt2JOGjatLz4u9UvvO3pLUxU2bohd1vR/FwmB/9b
mmtdtrP5s2N+/oZP/kBYETHNX60oaldO07yxog3V6XbRybCLmGi8wDrSwNeIMrQEurKA6vBl
epfMaLjGZKMGqAZAPy+Wz51uW7diJOHx8AfXoFcj0ClUHaGopxgiS93rQXnDcTGn7JSIqt/t
Or2v3BuepZThd+guP/bkDBqHbrBhHtVsjPp6YMD2Zis8gT4c+9DQYKwRewb829g9ZmY3cxhN
KwVypB+/RUSOho/DWm32a76xQOfDODh65noPoPqtMBOaoRyAm2qbFPztPhP946SAy6SWD37d
8k26vJ+l2vzMqFR+pOymYFFgwP3K0OFaKf+K2Xh5luhRVg+Ev7BI2ejBOFh6TtjX4ljTPw5m
1wnMD56tcGFBxMDdnEJl3ekCPPwcDBLZvacRIJjOUsEPvybcY04FVADSXM/jSkZpW9BLNR5b
1FTXIT7PMN6ueIhAdKDDHgYNN8up7ZE7ffZBByIXnXVil+Xt6CAXOV3YYRtegHBT3bl6SZsH
9atK6UX0PzT6LqVnUjZNAJfWnE7GSiwZL+E/32JXMkT68N1obDffi7Nyv1NqAmGqF31wWwH6
YV5JE/mHUxfInpG9UeNvGnLVgus6/O7X1b3H4/BglqQ9BhAz5l3rStt1tolpOpI+HErKKc5C
/vT958tgObTjY2LHSY3BdAYo85zLnwAE+Hw/ZA37NlnOl6YXhEjNI7SuAw7hcPqO9LPgoofk
DU+iGR28qpmfYDT8lg0/FKRqsufHvqRon1H8zlFF09Xpi0u87JnVSG7e1oqNBjrfLR3/5iR4
avcFaFAUpM+CGVX8lCL3UYEDa1wK29YxnkU/9JY5NIERb/FZagWO5yXv5gJ9+E9wWY44moU3
3P3Ov5x2+1bkhiAdmSUCUo/wU5eK7kntlpU/6c/lCN+VPlhht4neI+NQkmVZvbbgEk4KD9qA
tizPH4tdbAJCwRR9UMuNPdgwnObVmKCgwGZmZr4ZSDpDmdR7XdPOaJw6hiXxvr6xAVuvVNrH
LsiqdKop4jG7/DHOZOG30/LvyOutt9GktBDCnIPAE+7/JHK7Gn4FGbOt3VZxW5L5WWiRxzeP
urjt68U/sQYxJRMqopRPhpGnguFAw6Ye6wpyvSg6nkZjeADyNSYn+W66VR8IZafpnfGxFtnZ
GiE3UAxL7YvP7AeG2APYepCtTnNybFPHQPkHSRtn4NK5BenrRfGBT/+M4WdyYQ++4Kn5b8Ko
HeU2SqNOru0VvEGa10emd6YOe2c7NgJGNcp0OeCzYsNZxLoZJru3CMZ3+w5uyf2TEaCEKwCM
NYMTS7o+KFO7N1AdEG9Fz4HeIs4VeqOKFRf4+brBo8xK8AV/dT3aXxORz6TRR/wTNwKDuctw
03Tq4X+db+V6z/4nuyzSU/G3e7J0JtcuMgkP6txV2GVZPFMajwZckg0uFZcu5HTkXJsa8x/J
cLxtzYin5GqRaOHYgksJ93ab2vTW4lvEUlxt841mgICndnUa1NSyC9rw04mnaBSHBTFLjyKG
gYoeRkE9Lj/VCmmr1v5P9/AK2Ui5lGkV4ugsBOx6dqK9TuZ5uSaAa3fvgIVHofBKckafpHBM
AWFUNnv3RcSIrCx6IOduGDJr8b3JuQIxCLg1u4HaJuSCqsINhch8B/4NTz72YuTzDfh3cGHf
rcxYgp4FTq1lc76nC7q6fbxMaRZVfjqcbFCTUbBVtg6mUapmDh/Ao1VzXUMml1JdBRqJhBfN
kj8oJ0+U1wjcqPcTTCvHljuMmNtLET0AVdgxSqUA1q/qg3dazHJ5YBDKt34a5cmHQeXj8pXN
hOpSSb2uHlau3cfxPTRPqhfKnkDPKFPNaqTNCF1Pmh/xDzbCVNaKkPK+rLQW7p5VpA2cGCNB
eaKwVEccH+NUmTVyxe+jH6QZdQa8GuawW1JYo6VSJGJycqmyVWcvH7m9ZYQI7VObfvCb4D86
IvkeytKM31BkHsxEEraJuI50AYxIs+1ZX/bz37IRBLq9AivLPCFZ20crOPT+IS2v4Dn0YIfD
UH0lDfZ5b3e9QBsFVORp3o6Hw2WewktGlPHCfDrYsm/t0Dv368Hbq8HJCDdnw9syZCjD3mBT
2HQiHSbViOvOUnD3saFHZeONsuNM+9KXi/d1k1o3Y65Ud0wggEiMA0GCSqGSIb3DQEBAQUAA
DwAwggEKAoIBAQC+G9YX1BboIItyZdjEnPeoJXh5J3UQak+lkO+KirllYeNQML7ent3l9KOp
/IfxSDmVWtyYAUoxM3TGnDoJgCCn5jUQIdQkPXTxa7xeL11W4d0fqyaAOyMOXA2QZShE61dB
ly+bnIN1MrcwLe0q/TPuKS1EBp8LS/lQzW/PrHTnDHwoGinBZq58B/ErgmPh27QlkhamdXG4
dN2ayilmvdG06dMwrGKs2Tag5HKbUSMPA3BlKvvqrJhMb3w86xPf2MnvWzVWmi0tpE2zcCgX
4VtiikBW1QoFuFjiBWNJKWMXe2UKXYbdYFBPzaJiGCdiYXXJu1N16pSxAgMBAAE=
-----END PUBLIC KEY-----
```

which decodes as:

```
algorithm: AlgorithmIdentifier{id-Dilithium3-RSA}

subjectPublicKey: CompositePublicKey {
  SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: id-dilithium3_aes
      }
    subjectPublicKey: <dilithium key octet string>
    },
    SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: rsaEncryption
      parameters: NULL
      }
    subjectPublicKey: <RSA key octet string>
    }
  }
```

The corresponding explicit private key is as follows. Note that the PQ key comes from OpenQuantumSafe-openssl and is in the {privatekey || publickey} concatenated format. This may cause interoperability issues with some clients, and also makes the private keys appear larger than they would be if generated by a non-openssl client.

```
-----BEGIN PRIVATE KEY-----
MIIcOAIBADAMBgpghkgBhvprUAUCBIIcIzCCHB8wghdaAgEAMA0GCysGAQQBAoILCwYFBIIX
CF0APsq9Omp8nro3N6w2Q6IcFdZwXkViz2YG6C04cBSuXzreoI2nq7/EVjqV0dVYtWemAk+T
dZpvQmMPkjQTbD3f4xBwlnQQGwyIaqV3uLMaSixrNDRQfjmbMzkJveKu+GIDMDNRhhYYFBQA
BYWIXeEh1cRJjI3MkQhQTUmgHNzFTYUVgdXMVc3MHCERRREdWBXhCcwUBBCQGR4cRBghAVXd
NhJFOEYHVVVzNDQ1NYBYcBhoeBdWNCB2FxMkIGEhBiFzUCURA0FAUDMUczFmhhc2MgBQMnhH
hRyEjRAcEByZ1FDAwWFFhdVcoZXI0R4N0UiRmdDd1NAFkYSJIRHgwgjJ1hwJ0N2NXFTZ0EjI
Y4ghSFcAFngxMAIiBQJ2chBmhXgnYDQkiGZTQwZAGDc2YxY1hUUEMBR3MViAVjJHhQdmFgdE
icYRGaHOBVDAFNFh4JWFXEWUHhidHU3h1aEMkBoJCEnSGSDETR2QYBBQWUYdCOCMwCGKIZkY
gnGGMRaAQWInAnVYiBSHEHYQCIUGcEMGRFMzhyEQBgMTAoYVBlhYgDRVhSJkUSgohxGCMSZG
FVBJEhwMocFIGNUY2IicDYnR4BiYUB1KDREMlRYNnczNyQQFWZCQVEwBHYASDAFYRdVYjNDN
ICcEgUY4Y3A4A0FEZGZiUjFTUzAkQEUTg4JFNVRBU4RjdYcAZ1BCgTNGZAYDIydwJnSBVlZx
HM2VUZwJGBDNlWFFYZ3gGZEUhBIcGEnOGNgdQEmFgM2AjhDWIUGdVYmMRBzYCZjhoMEiAI0c
NkYwVjRwcgRyR2E0BXEiEEVRcih1KAU1dScTYmg1hXdUBQFocFBUCFCHZyQhVQYIBghAQRSH
CiCVVhoFEAEV3UnUCIzaHc2Z3QYMBIkQEAzJnMxd0EkCCWFM0VHAYgWBYMGCHQCRjVoGGdTC
aAiGdDdgcCFVdYiBQSUSZwNxBoYRaDIwJEAHBYSAiGRUUxYEUiZoYABCQGMkAHMjFBJlBFQB
RAERCYHICOGYAFUUSB3hRgzKAFiNFJGGAdFGEhlVgImchV1cIcUZChTYHg4hXEkFoQDgRhgU
MSMwgEMyYYAAAgVHN3FxBDgHQkCCNHNwZzVFWBMyQQg1AjYVN2FhY0hGMjdlU1VFMnZEN3MC
FhzBIEwABN4gXJmRhRDdQgkVHUlFHKAJBQFBGIGVmICcWhAcRYYVRVECGRXQFYjUiMYOEgwd
MEdERGZIdSADhIJQdDNHcVVxiHNjUVQwUUAXKHFxWDOIVzQQBQN2iFQUECQEgGGGh2dRKARQ
3JCZihyAnRSElQoIXghZmdgV2U1EoUQdGKFZFZBcHJAVSh4UHBYJyNBR4AHJXMnAIhABmM2F
RhUjRHgSEygXQkAXUHaDA4JCNYYlUVF1GHNAiFQDQyQINCNzISI1cichgTYVJTdEAISCExFh
TJ4B4EDI0EVZhJTREEBQFhGGHcmhiSCMoVEEUYkgAcHQQRhBCF1N3AYaFQoJjQBCFRSQQRgg
E2EnNQUnMgVghRSHgjAoAyMEQoElVWdXdXRiEEAhI1UyMlRBJHeDAxhyIyRWNBMTFWaGdwUU
zQzM1QIBmRoNVR1R0EjBFZzUxURKIIFAUgRWCFkBjVENxElIEZ2EWAASIERYjCBV3ZYR2RYJ
h1BCFzYQiCRmNCJ1EHR1VnhYVBZgQ2ZyASSDZlY0BTiFdodmgXEjAFQFFjNnUAUVYhQTOBgC
XUiJjJDUUZmYCFSAgVigVcIg4IGcoI3RWEnB3GCIDJYSGKAZiMyUXBXh2h0IHAiZwQSU1UlZ
YmBYFgJIEnUkV4BzNXN3cGQxEVc1coVgUkVSBGiHIHdCQgcxBoJmhxM1ZAIgMlFQd2MCgxQm
gYXIGOBNyQWRnZlKIE0NjGDIYVxFSGGAmSENSdYKCE1gTclUXU4MDAAhzNoSV/oB1DjWBLZ5
QVd2ppLs7YI7BBTrICpkXQcPWX0jFXuibD+6+YL56TvIdb5Cn4kBeWVtJTIIuRyJVK7NbzKP
Njo49ytpj+GP8QwnfLuDlYY0v012JLonxvrYc1izUZbau5t1XU4u/FgOvXpP/HuQKxwBefhu
xZedCArFTiDa7pPKjK5TJPCuCWvpz0hOjqu01KwvD0EUUIUq7SmvWyt593X5XwPx8VSYaFqL
cHNmFvGYAGT2MyBlkmsrZPjbxzeqF+Riias6fR/DT0CGHK/aB5BsKFgup9kEOnVEhDRlRHqp
WndVHV3PBvUBmyAJCOCshWzu/Irx+HzEU5raRqzs/5wwTZcZzahPx9Uo/4FeKWkw/UcjI+ux
0tj0mZ5lLttGwfC2JuH+G2DHUcorccCwFnaoHJ7x9jf/cnKcOMFji4UE5CHVKnAEgwWU9neD
V4JAY6nTKH6nZ953w2u4GMI/pvA8RYaylEhoHmHTSvH7xbFlAH0FkI6f3DuGD2UVMkk6+EGu
xp0gws1X8ZsKSECm+0/yWwtRmbyVNW0js/8thly5eet8ZaV4AG6MysyeW5Fp3CvS0AKBfzfO
YZAxzEi1OjtNtuYiLmXrvFm6xEewIQLBix7M907KuOY8wuRNIArJKzrGMybqjnbk2woBNFJV
gZel8xMX0dWvcAbBm0K1wxv/n17nFZ/rP2BqGJyA6nGu7m+vREiUdahK9YU4xAWGwjb0oeHC
kWi6RPEGhVgWmlcVkUGBvWHQYdiSTu5plMAPBeWVZneGwbVZ3B0ksLXfGIc5lTcReSGyeCVK
mE/GtH92lz0fC9GQWgdyTK4H23xP5PlRGlzVLJ5CVA/rucNg6F7AU5bycpC557ugsdO+UOLh
XTl5RKvO4uxrqZw4OOzKXR+aZFaY7AvpPSIXYQRxk4dHAW5ra8MpI0b8aAJGf4fE4ad2w6rg
K7NUIUOfpBhX9QxcprPyXsp7AfQsoCBs2ipf7QoJNu6GkFCFijTFxBAt2rK58QoeuxYVjQid
c1AtYhBFyY9OronGTsaEtpL8goTptIX10f5YYmjmTd4FgB0Iwiww1Re7/G1eiRARzp4M1u1V
xp+1pRZn8mpx0PIDMSfRieoicAcXhjMgCCwhxf79RWlxUMpKU3xFNVN8kcS1XXy1OUt6myAp
2SzUEJT0l/zT9ni/JhDhAb9vd09ab9DWGifhIyOkoq2Cg5AlTQXAwI3kDp31gg0cDP1O1j0D
pzmFlo1UT8MauxEH4h1AHWvyM9Eiul90Mkp6ofd1gcFC3J4fSX4y8Ut4n4vyr00k1UVKthLI
h+lDi6KZBl6v2U4GwLyU5bx8O5j/iUyims+sxjlsKap4YWLK1Z1EMnZUwLyKgeUBljTRokTE
1oMahgoV5W5FjNV//mTsLWQvQWKelX5qpCvzTXIOS9T5Ucxex0/7lFn1/k/I1Qkr4FQgtDze
```

```
rppntEHXj6qcpypsQwPZ1aGwBNvbAlwbX7nt2Vz/twx7kVInyvB8gROkG5I+SAPiTLCsritC
oyeyEXvXPm3SbU9bgCrqxZ0uDue0BrfdZVagEpL0Lbr+TSrVVNLv/DBfQsYD+DmMF1ZP5BkT
l9eHmEReQCIiA4kx4GSjtDpcQNil2m62jlM5yVqa5mpDKo7VvOMAM/FZehEWqtEDDwP4KC9A
kVhpt3/4ZoT5HX1nxaBlfm2Ts9i0VA7SJxY0HRRI72mcFbxlm2bvw+ntiX0La+UUblSZlafu
xDYAC9f9oAssAkaMgbdgi+GNQh4msBMI2SEDdC9iOTrvbx1oFlySi3StSKND4iAxs6c4IxY/
THVCPd06SFyrqYW8A+Ex/tGfeKX8rfZVnkMaNScMpxytXd16nlvDxWGT9RvNuDLY5CzQwz+n
OirCkx655apuXisz6ELRFvxb6MI0FkdlO87MuYQ7Q9uGxZE1QcRAvIoqTzl+E4IoALr0sf69
V4m3RMvALIPsCJVrXou7SUUKEwhozYXyVe83OQZpqeQsB57AiCA+YXAv20Fgp5xJQerU90xu
8p0zhIizEDiRWGt68IDeqPjcvU+0DWxEyHqUvBZe2vmNCpV4CFodMleMZ3LzP9pBUM4kB5Co
Wzz9EKgPAZD8gYUZn2bSxvX8zsBEkW0D1RAZMsEex8c2Hb4jJiiEqwhyLzysagU8nFdNZ+W5
Dlk7k3E9OSh7PuBKCRZXd7cVym0NHTEMHjOaiPHsgUWB3oC+D9e088DUEYe4cu1Q36Bg11Vg
YRwBgnt2OkWzNKRFSN2mPp4fUY/3bPWVqOGQZcclhzVSNamMcpGWTjtWfb6utfAktCR+fhY9
X0jsn0D29PfLndlw3syxoi9gAwy3gFCvlfTbY+n7AbmEO+R7CKOXywqcH1vIUfHtWUaPqBk0
rfviLXewdS9xu401eIGaTZlWIePIwE1pK0nedALw1ABGnUfFm7kz6UDRA59X1BXb5oAXzzDY
IWPV84n0XVFaFW94glkwO1UtUtTioUUTaSAXvh6YR0kxMhByHkDHpQ+B7dN25hAc3rYA5Nul
bIyNimLOxyCx3v+ooRtUzmseT8rvS1dBdHEvt5AbENG8oFWhyquwN2ftEDGRqJs927RxvWBO
kkak7vyiawuFqU9iMUyQ6Pcp+tOvJ+m493C5H3s8L49gfiBiqr1hYZuFqO9u87wTuVt0eGF0
EfOKKdiWj/+cvEID4XRE5ay66/F4rHneLn6Ucc82Z1Ga1BInqVUdD4O+4mGt8z9FE7p2JkFc
uKFxvFtsVq4kyYg7S/t2km7HkIU4Byqa6EGJPHbxdOX/0gADEvovhNHy1WlZEpinPEKTP+ji
T5IeA95LI0GP2sv3a9ZX4JUKFq+pAB5t7EwcGj9/htQ1MOKItwAGgYJvTVKmiCgO48Uxqinq
qyDKcXNC9cqw4SVVOa9ulu7O0EgARObu0mxA1PMvJZ8VPaARQl/vVejgnLUUqoqD0zaZprt8
uUbkSO1Yr3PUz5YI6ldIuutuplG842+FfW6oFU/cfsD2Rp0Yxq8O/gzOKMR7KFhhi7sLKs4x
RwI1Rw2cUtXQz/5SH936SDtdgoFTmeugaa/H+O2hIbUJGQ2IS7NT3NG72HhNXe0qCq6v7ZRM
SaDL62c5RwhpxVJrqGpr1u+cYupZ+YAZfgaNH/ePo8BZRDgurCtSS0QH82/2UUfahWsPVEiU
khmVheFRhKqGD7KvTpqfJ66NzesNkOiHBXWcF5FYs9mBugtOHAUrl86Ns8l24jV6Ut+TjYd8
UClNoWhGe/v2W/gZ34NQGlUahMxLY68nqH2BXO5bjMHbE4pGGGuNejGJnJoe+1/kd9+Mym6L
pAZCKRtgka9wXR3i/MOC+qp9OHi2tt9cpur11mk6NPWjdVMqXxOBYgeWESR5k8fYS6ttRNPE
Q5ZtNMV1srdov+mjWRiTmGIXWUwXWEI/5LQAthtaSywFPHj96+kLlm79lbYeaVfSN/URojQZ
2gx0DFK8m5ZA9GaRQZSL3eavfzw2Kd6pOdFNemTquM/i7uGFc4/Tt2JOGjatLz4u9UvvO3pL
2bohd1vR/FwmB/9bgwLmmtdtrP5s2N+/oZP/kBYETHNX60oaldO07yxog3V6XbRybCLmGi8w
wNeIMrQEurKA6vBl+6OepfMaLjGZKMGqAZAPy+Wz51uW7diJOHx8AfXoFcj0ClUHaGopxgiS
QXnDcTGn7JSIqt/tW+TOr2v3BuepZThd+guP/bkDBqHbrBhHtVsjPp6YMD2Zis8gT4c+9DQY
ewb829g9ZmY3cxhNj6hKwVypB+/RUSOho/DWm32a76xQOfDODh65noPoPqtMBOaoRyAm2qbF
PhP946SAy6SWD37dZ+t8k26vJ+l2vzMqFR+pOymYFFgwP3K0OFaKf+K2Xh5luhRVg+Ev7BI2
OFh6TtjX4ljTPw5mNQ41wnMD56tcGFBxMDdnEJl3ekCPPwcDBLZvacRIJjOUsEPvybcY04FV
XM/jSkZpW9BLNR5brET1FTXIT7PMN6ueIhAdKDDHgYNN8up7ZE7ffZBByIXnXVil+Xt6CAXO
YRtegHBT3bl6SZsHxfE9atK6UX0PzT6LqVnUjZNAJfWnE7GSiwZL+E/32JXMkT68N1obDffi
v1NqAmGqF31wWwH69+EYV5JE/mHUxfInpG9UeNvGnLVgus6/O7X1b3H4/BglqQ9BhAz5l3rS
tolpOpI+HErKKc5CXfo/vT958tgObTjY2LHSY3BdAYo85zLnwAE+Hw/ZA37NlnOl6YXhEjNI
Aw7hcPqO9LPgoofk7oADU+iGR28qpmfYDT8lg0/FKRqsufHvqRon1H8zlFF09Xpi0u87JnVS
1oqNBjrfLR3/5iR4qP0avcFaFAUpM+CGVX8lCL3UYEDa1wK29YxnkU/9JY5NIERb/FZagWO5
5gJ9+E9wWY44moU3vJY3P3Ov5x2+1bkhiAdmSUCUo/wU5eK7kntlpU/6c/lCN+VPlhht4neI
kmVZvbbgEk4KD9qA+FetizPH4tdbAJCwRR9UMuNPdgwnObVmKCgwGZmZr4ZSDpDmdR7XdPOa
hiXxvr6xAVuvVNrHfcVLsiqdKop4jG7/DHOZOG30/LvyOutt9GktBDCnIPAE+7/JHK7Gn4FG
3VZxW5L5WWiRxzePYX4urjt68U/sQYxJRMqopRPhpGnguFAw6Ye6wpyvSg6nkZjeADyNSYn+
VR8IZafpnfGxFtnZRPgGiE3UAxL7YvP7AeG2APYepCtTnNybFPHQPkHSRtn4NK5BenrRfGBT
4WdyYQ++4Kn5b8KoDr2HeU2SqNOru0VvEGa10emd6YOe2c7NgJGNcp0OeCzYsNZxLoZJru3C
+w5uyf2TEaCEKwCM0+FNYMTS7o+KFO7N1AdEG9Fz4HeIs4VeqOKFRf4+brBo8xK8AV/dT3aX
z6TRR/wTNwKDuctwJQv03Tq4X+db+V6z/4nuyzSU/G3e7J0JtcuMgkP6txV2GVZPFMajwZck
```

FZcu5HTkXJsa8x/JWRCcLxtzYin5GqRaOHYgksJ93ab2vTW4lvEUlxt841mgICndnUa1NSyC
04mnaBSHBTFLjyKG1FOgYoeRkE9Lj/VCmmr1v5P9/AK2Ui5lGkV4ugsBOx6dqK9TuZ5uSaAa
gIVHofBKckafpHBMpcdAWFUNnv3RcSIrCx6IOduGDJr8b3JuQIxCLg1u4HaJuSCqsINhch8B
Tz72YuTzDfh3cGHfvmFrcxYgp4FTq1lc76nC7q6fbxMaRZVfjqcbFCTUbBVtg6mUapmDh/Ao
XUMml1JdBRqJhBfNcKxkj8oJ0+U1wjcqPcTTCvHljuMmNtLET0AVdgxSqUA1q/qg3dazHJ5Y
t34a5cmHQeXj8pXNalHhOpSSb2uHlau3cfxPTRPqhfKnkDPKFPNaqTNCF1Pmh/xDzbCVNaKk
rLQW7p5VpA2cGCNBj38eaKwVEccH+NUmTVyxe+jH6QZdQa8GuawW1JYo6VSJGJycqmyVWcvH
ZYQI7VObfvCb4D861QWIvkeytKM31BkHsxEEraJuI50AYxIs+1ZX/bz37IRBLq9AivLPCFZ2
OPT+IS2v4Dn0YIfDztxUH0lDfZ5b3e9QBsFVORp3o6Hw2WewktGlPHCfDrYsm/t0Dv368Hbq
CDdnw9syZCjD3mBTnYh2HQiHSbViOvOUnD3saFHZeONsuNM+9KXi/d1k1o3Y65Ud0wggS9Ag
A0GCSqGSIb3DQEBAQUABIIEpzCCBKMCAQACggEBAL4b1hfUFuggi3Jl2MSc96gleHkndRBqT
74qKuWVh41Awvt6e3eX0o6nS0z8h/FIOZVa3JgBSjEzdMacOgmAIKfmNRAh1CQ9dPFrvF4vX
3R+rJoA7Iw5cDZBlKETrV0ETP2XL5ucg3UytzAt7Sr9M+4pLUQGnwtL+VDNb8+sdOcMfCgaK
rnwH8SuCY+HbtCWSFqZ1cbgrEN03ZrKKWa90bTp0zCsYqzZNqDkcptRIw8DcGUq++qsmExvf
E9/Yye9bNVaaLS2kTbNwKBeRf7hW2KKQFbVCgW4WOIFY0kpYxd7ZQpdht1gUE/NomIYJ2Jhd
U3XqlLECAwEAAQKCAQAyhSM34dzUgxGLrRUV6sDFpm+Fgr7RRe80iHef0Y3DK2hE/y856e3+
IDEanGFj9VWYIzVMD8uvl4UI4qtpqusCs2KWjubZWpuhLIg6X0vmss8Yg1sP6KdAQaY5ISi6
gEwVd//m0oj8tCWKYCoOqosKV1b4JOpPDjmLB40PwT5T0B10IhZLfgrAwKofbT+bpyuNbfZv
oxUvT9LYWtcF4aPdA7GqgwKAybHq0UYzc2Uggki4gKtlw9MHdzz0u1jnu42sJ7qjadVlWH/U
uK5jmzcHmWFbzrFP2O6p0Jpq0AP/EcNKjbD0pxUuGF5RbbIuTQeXfG3z5pkhAoGBAO7qVBwU
Q00QmZ46ZtvfSUYslFwrQhM21zVUJbV4tfPZYlK7cInQvlwxy3prCdpbROMcb78eL9aFCjaS
A3e4ufpm6Ncu5a/Y2W5ScSw1N7EZHeIGj3rZu51b0WZZakUHupas4S7yApQT0cyg4Fru8yr+
24bwhxEWl6nAoGBAMu0CUB8+lZih2vDy9sDm++DTlDJZDKPfOCCj5IH6YqtdkUm/+TUmuINZ
yowUulwpunPDFs3W6aGr6XnOYkcbbQ5dHbhuF9i+JY84rW/zhLLd/nGf9zx1rKanWD5B1zC8
yM0B7WY1A0Rg2D9K8OJ+FTwEAHq2nkHpVmPTnAoGAGuOks8RXwWqjXHg2D4adYSb6pn52KXF
iFM5zsAj18Yv11PnjoJ8tzZKNJCoH4duD1UvOB+SN3cb7b0j30KAoWjZBaWvbwiIMVJmplQB
0i+oXwuiaHWmEBCHhX3OQ0lQcW/j5Hx99ysywQCebGKDsHVqzSXDj1xi1zmDuECgYBiwE236
oE5cLNg3vaEr1LDzsx4S8MKuKD0noxRRuVPbpFNrgLHxImP+Z3WhwS6zHTuZgRseALDUQn32
89IvC6dtNnHmNBmk47FYQLrLG7525Qb5engFr7TZ3P/3tT7zMwj8cZG/+bUywewzisYKVus+
DdyJze4X48QwKBgQC5Q8YDafkPhO0/riXfHE4L5NnPLWX6t3TINt+eYYnF47Hph1o+TQTCSF
BSZLFwpH1i/y1P/+z2rOuqrAuSkK462Uz6u7mslFVXb82baLM1Kf0h48Y+YuEJRlRVnNCpRH
xo0q1Hy6tHThjXl2WXxTQEKyA7olA+T2uI6x1Q==
-----END PRIVATE KEY-----

B.2.4. id-Falcon512-ECDSA-P256

This example uses the following OID as definid in Open Quantum Safe, which correspond to NIST Round3 candidates:

https://github.com/open-quantum-safe/oqs-provider/blob/main/ALGORITHMS.md

id-falcon512 1.3.9999.3.1

A Falcon512-ECDSA-P256 public key:

```
-----BEGIN PUBLIC KEY-----
MIIEBTAMBgpghkgBhvprUAUDA4ID8wAwggPuMIIDjzAHBgUrzg8DAQOCA4IACZdJCeFxldpf
HAR1VY2ntR4Rc6Q9UvTsy6vo7xDbFSewjE1Ei4gmUTWz6PfkpoDkrOVqA4tkTvQZE23jQKVf
DziGZuOtsw2aCjXlJsIdJieTytFvEegdOsagJJLd3IjqQOu41LCiLGXLQ+widcNBcZ/Bs1oC
qJtjofEiRcUCie6vnMiMiLsMLl8S8MBn3CIDMB/319AmgQro0YqMjEsS3VBzIyCnkmwy3QAP
oJkrrBDUqtLkXyh5vaeirZk9DnFDJc6Yq9Fz5yHVoE120ZjVyJBC0O8p2BX2YXm3YhOXWoDg
gJhRlYHnUg28oMpaBav7V+0fyNAalD3dpLHGVqAVblvY6uDSLv1uBY9DcHMhPrHDBjaE1Mbk
wGrHQh4AM5p5pFwcAGSIKYZsiUmPWctwItvMV+ydsdQvEyISfBq+1FyLdTFqsQv4GfUlpk3M
cvNVdSD2FjHgLOqkZZyqspZil4jCNRCUW64cSOOT7m0I6nC2U+s4Lcbh6yuu0z8mBcj0nWud
WICEjviAo8dkevJMU63lOxbAXXe/ZSgJ8ZDhL4dXfSXLhFeg07aOmKFUah5a8i5JrFSnh6zk
nf1GyfdnDBiAcsYlxtFEjYkG8cTJIOwhRslem6TAJhtxSM+cHUKIQY+Dl6QmNFKR1wBGqap0
WAtu4VK2rTSL6ENRmBB+pwStWl0+rAVjRtONLkCX7aSfh10yaaJBloTJ1NLqQI2aTBM0Ecjh
PBCgtf4+2bQTFLe0pKcW2aB7TvMHcUKyHQFMu5erQXbY1FmP1w2y5R+GK1myuJIQMrWCKZxd
nk5EaunOYxIGgslmINGjmKmzwTiuIwsF2oimRdMUla36Ss4nmcCzlONcQ1NKtdvXaWJotbGI
xCFUvspsIdZemJ8MZxZIAotW5u12IaIgmiMdTUoH1O5mpjso5iUhynqlOqfw6aJbJH3+FxMT
N0UdzR65uiMfSiq/HVb0zhB0gJLrqpPAXaMgiYZHBqQqeQeB5pwTYM5xuiKIbW9KG+gBIOCi
mTUd5U+4hceULMCpaTff2636q/ofEAZlrfQC2aV/GAWx6iOm67AQMVTKG34XFqVY+C9eLyip
FLAQt8WaxSAuEScapj1+xpCpUlUhm8Ca2AbvfEi9pit9Rj8y9t7TIpDmMFkwEwYHKoZIzj0C
IKoZIzj0DAQcDQgAEFFnVIrPg3W3QEt95Z4WQn1cmoZV9tpo20FG9umA2B7CSaf7q9FwgVY7
HymrF59in1pXEPnEWLZ9xLO5A6cg==
-----END PUBLIC KEY-----
```

which decodes as:

```
algorithm: AlgorithmIdentifier{id-Falcon512-ECDSA-P256}

subjectPublicKey: CompositePublicKey {
  SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: id-falcon512
      }
    subjectPublicKey: <falcon key octet string>
    },
    SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: ecPublicKey
      parameters: prime256v1
      }
    subjectPublicKey: <ec key octet string>
    }
  }
```

The corresponding explicit private key is as follows. Note that the PQ key comes from OpenQuantumSafe-openssl and is in the {privatekey || publickey} concatenated format. This may cause interoperability issues with some clients, and also makes the private keys appear larger than they would be if generated by a non-openssl client.

```
-----BEGIN PRIVATE KEY-----
MIII9gIBADAMBgpghkgBhvprUAUDBIII4TCCCN0wggiWAgEAMAcGBSvODwMBBIIIhgSCCIJZ
+0I+BBBHBEB9A//69D3BA65EC9A/A//D/E/G5666D5+EA7787H/BE+CD868BJB8DDCC/+C98
A669+A/93GB+/HB4AEB77+/D/4F+CF5+B86B5FA6AC+A7//6/+D+7CE9/GAAE8CE/DDD4+FK
AD489BA/6/AADC4C9B7B9//EA+FBE7BAFAAD9/BBC4G798/E6D/EB+FHCFDAABEDABB4BCAJ
CA46GBBB/8F9++C99+D4DD6F+AECAEBDDDE+A++GBD9+AEC9C/5/9A+D/EBF+DE/B/A48/E7
/8A++495CCDABENGB37C6D5+92C9F8/D9HDH/AB76CFAA+CIAGBEJGA3JFB68BGBAAC6HDHC
++8/9EHBE/BECF+7/5/CE7FD4+CD99BHB/w+/9BLB98/ACJ9FDB+/EA+CCDA+8988F+E8A8B
D9/5995AAG//88+BHB9C+2B7GCH9CD56F956EACBDEB+AEDGL/D+6IB9GBFAD6CF6F7D6KD9
DDEC/CED8E97B+45FCF1z6CDD9DBJFCA9/A+CCFDKG/+EA+A7BA79KABB7717C5/CB//+FC+
E9C+7H7+CF/358GC//FBB46/+F/4/++B+A/B6G9DFB6G/8CE9B3+7C/D+FABB9/EDFAAEF+D
88D7CDEFAI2HG87BGB/C+/A/ACDC/AFF/3/8F/E/D966B+97DBDB9EA/9D/FD8++8/B/AC5/
C9HDE+5DA/CDBFB1AC8FBFE/+/DA9AD/+85A+CE+D6+BB9/H5C+CDFCCE7+B/GD6+C/CEE8G
AEC/F6ADA6CF9B+C8D+D+B+K+4ABAAA4/8CDE/A+FBD/9I/A79E/8++C/+AB79D2BE5F4C9+
E/5+ID/A/AI//J/A+D5F8CA+9/EEEA+6DD98B4DHC8CDBI/C8BCC5IK+9+7CAFBH9BH3ABCD
C9AGDD+8CC8EADBC/94F5B+6F8++EK6G8879BEECFCABC+6OvxsQz3+SHpzfP35AjW/Cfp9g
P776gMtBxMJ3woSCuzpJi0p2dnrByfhDSreIBELDycfAgAo7AACKfYQ97kPBBMW9iYzOdNIG
FuzxFQsWuy4qQxgDLBYe1AIBLPIqFgPj/fFo9vwDDO4W+vnq1fP68hkZCSDl/yfpGhTvHhHo
b+STpCf0S5wMhyvIR2/L75BcC7xozCxH88fXHBy/1ChPZ+fsP/hsF+djx/+MKBesPzwP/+hc
8MFxX53hwAKwYfPOkuLfjUBQfnId4IDeTfIUFkHhox+u8l9QUq4vgVBff1EwjVLhsC6w/z9h
xESFeoHBewGCg8XtdDnCRjo0fcF0yDTEv4uzxYA0BT8++gIAQ708d/z79nS5/YO5fYi7RwJ9
BOLsANLV3hP/Af/zDvS71RLv1dP25t2wGwfdNesxEgIlKgPzAfr85rAWDAkG7f4n+hMA+wzu
RDwoVADYvBjoz6gjmAwcV2gf+4AT5BSgCAun97gX2+Qq39BUT39nx3/DAtxAZ3PgOCg8MKfA
n+/eMHD9TuHu365OQE9/0BA87qyvX36BMOFB39OfXx/AsC0fb+39EJCuUS3BH75ub38Or6Hu
9wFAvK7BBYQFf0TGxAB6P/7GvwUBxMVA+sZJDEHMjwH9e8J2BFO1AMJl0kJ4XGV2l+6WIcBH
ae1HhFzpD1S9OzLq+jvENsVJ7CMTUSLiCZRNbPo9+SmgOSs5WoDi2RO9BkTbeNApV+CUcPOI
62zDZoKNeUmwh0mJ5PK0W8R6B06xqAkkt3ciOpA67jUsKIsZctD7CJ1w0Fxn8GzWgIA3Com2
SJFxQKJ7q+cyIyIuwwuXxLwwGfcIgMwH/fX0CaBCujRioyMSxLdUHMjIKeSbDLdAA836agmS
NSq0uRfKHm9p6KtmT0OcUMlzpir0XPnIdWgTXbRmNXIkELQ7ynYFfZhebdiE5dagOCje+AmF
edSDbygyloFq/tX7R/I0BqUPd2kscZWoBVuW9jq4NIu/W4Fj0NwcyE+scMGNoTUxuQFmnAas
gAzmnmkXBwAZIgphmyJSY9Zy3Ai28xX7J2x1C8TIhJ8Gr7UXIt1MWqxC/gZ9SWmTcyVl5y81
PYWMeAs6qRlnKqylmKXiMI1EJRbrhxI45PubQjqcLZT6zgtxuHrK67TPyYFyPSda53UTVYgI
ICjx2R68kxTreU7FsBdd79lKAnxkOEvh1d9JcuEV6DTto6YoVRqHlryLkmsVKeHrOS78ud/U
2cMGIByxiXG0USNiQbxxMkg7CFGyV6bpMAmG3FIz5wdQohBj4OXpCY0UpHXAEapqnTD4BYC2
ratNIvoQ1GYEH6nBK1aXT6sBWNG040uQJftpJ+HXTJpokGWhMnU0upAjZpMEzQRyOGGYg8EK
j7ZtBMUt7SkpxbZoHtO8wdxQrIdAUy7l6tBdtjUWY/XDbLlH4YrWbK4khAytYIpnF3vh2eTk
c5jEgaCyWYg0aOYqbPBOK4jCwXaiKZF0xSVrfpKzieZwLOU41xDU0q129dpYmi1sYggazEIV
mwh1l6YnwxnFkgCi1bm7XYhoiCaIx1NSgfU7mamOyjmJSHKeqU6p/Dpolskff4XExPzk83RR
rm6Ix9KKr8dVvTOEHSAkuuqk8BdoyCJhkcGpCp5B4HmnBNgznG6Iohtb0ob6AEg4KKZUaZNR
7iFx5QswKlpN9/brfqr+h8QBmWt9ALZpX8YBbHqI6brsBAxVMobfhcWpVj4L14vKKnA+oUsB
ZrFIC4RJxqmPX7GkKlSVSGbwJrYBu98SL2mK31GPzL23tMikOYwQQIBADATBgcqhkjOPQIBB
hkjOPQMBBwQnMCUCAQEEICXQxbtuhGjYSkyLz+K0V0tRyxgEBlQcElOVwHZSA4QL
-----END PRIVATE KEY-----
```

B.2.5. id-Falcon512-Ed25519

TODO

B.2.6. id-SPHINCSsha256256frobust-ECDSA-P256

This example uses the following OID as definid in Open Quantum Safe:

https://github.com/open-quantum-safe/oqs-provider/blob/main/ALGORITHMS.md

```
id-sphincssha256256frobust 1.3.9999.6.6.1
```

A SPHINCSsha256256frobust-ECDSA-P256 public key:

```
-----BEGIN PUBLIC KEY-----
MIG/MAwGCmCGSAGG+mtQBQcDga4AMIGqME0wCAYGK84PBgYBA0EA6HRU4f2vmr2LV5vZVlan
Ly8ZCfheVqolJGrY5GxpNwvIt8fK6swNtftSgmrC+fCDE48/fbzX7a2U3F1/S3TBZMBMGByq
49AgEGCCqGSM49AwEHA0IABFjKamMP3nn7Ua8Y8XEJtqnp7ya+Ino3UoxjMhhVKHx0fQxAz7
Eytrtq3H7e59JYdkceK1h+T8jZFyUP5e0M=
-----END PUBLIC KEY-----
```

which decodes as:

```
algorithm: AlgorithmIdentifier{id-Dilithium3-ECDSA-P256}

subjectPublicKey: CompositePublicKey {
  SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: id-sphincssha256256frobust
      }
    subjectPublicKey: <sphincs key octet string>
    },
    SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: ecPublicKey
      parameters: prime256v1
      }
    subjectPublicKey: <ec octet string>
    }
  }
```

The corresponding explicit private key is as follows. Note that the PQ key comes from OpenQuantumSafe-openssl and is in the {privatekey || publickey} concatenated format. This may cause interoperability issues with some clients, and also makes the private keys appear larger than they would be if generated by a non-openssl client.

```
-----BEGIN PRIVATE KEY-----
MIIBMgIBADAMBgpghkgBhvprUAUHBIIBHTCCARkwgdMCAQAwCAYGK84PBgYBBIHDBIHA0PwP
Ulg3VLrZC7cGLqF0jRZrREj/l4kKF4JsLTjRR2P4RLqEm0qBa7ukb4ytHE6HDfM0h6dJ19F0
SO6Oh0VOH9r5q9i1eb2VZWp4rYi8vGQn4XlaqJSRq2ORsaTcLyLfHyurMDbX7UoJqwvnwgxO
281+2tlNxdf0t3odFTh/a+avYtXm9lWVqeK2IvLxkJ+F5WqiUkatjkbGk3C8i3x8rqzA21+1
sL58IMTjz99vNftrZTcXX9LdMEECAQAwEwYHKoZIzj0CAQYIKoZIzj0DAQcEJzAlAgEBBCAw
KKsZbXlaZBph1ixcUhlNiZ1qp4LnA90Nm/rArZw==
-----END PRIVATE KEY-----
```

B.2.7. id-Dilithium5-Falcon1024-ECDSA-P521

This example uses the following OID as definid in Open Quantum Safe:

https://github.com/open-quantum-safe/oqs-provider/blob/main/ALGORITHMS.md

```
id-dilithium5_aes 1.3.6.1.4.1.2.267.11.8.7
id-falcon1024 1.3.9999.3.4
```

A Dilithium5-Falcon1024-ECDSA-P521 public key:

```
-----BEGIN PUBLIC KEY-----
MIISADAMBgpghkgBhvprUAUFA4IR7gAwghHpMIIKNDANBgsrBgEEAQKCCwsIBwOCCiEAk2cZ
zZ6OzKssprE69H3WTaYIC1McjkUgD3uvSORYQwbUPEZFDach1iOJKkiXpb2nl91g0CJJzF2e
t5WK4jeS+vE+0iAGyRERTgO1V+XOAdbL/oXaltcsw2e9eaVkpXxeq7r13LHc2myoarAbUPhZ
0okF432s0RLmfacYE/uwj3TlrIHE17ELG7r75Vh5dLx+m4mDDXSpmdBe7z9jb7s3UIsxvsLp
ddj8QOMY+Jq8YMvEKYWoNh/4+IMhkKQo1jFOuArFYBRmEZPp+CZDjzVlvEQQqDOCIriRRwqR
0oRtpfws2gHHZy8VOqHZAtGdNhWhUvGo6xfcWAdFYxQIcNKHzZTlPh0MZcCrzd25nG0hgUe7
kHV526G5eYLLThmUY6b2pLH+LQi15szrUHAl5pch5KYgQ0S55Ya5cTotRvet+Ej7URTrpERu
5QftfB9lt/mcWzTQMEZyhH8MvTP9PayNbo1DbmDDLYnX7ts2sG+tC64hWNsMQQZ9oqXA1++V
RES4NxJeXDRRCUE7rR37XD6WJWa8bnzEix46lSIT2YtRwVqKsEIzpjHs1BuwoMOAsVa15iLi
7Cxw2r4RcAH9NviaKhS2elGu4cr9DpJhD4+bMi6pJ/jVU849nhfpfogXLPB/2HbHKdd3+6fs
BfiPrIu004g9QPNdKMavtUA/l2gqpw7Qz5cgry6rpK47s2zicHYvEjHtIKCu72/9stEU5qYQ
j2cfLzJ8WcpKFNnSZu1O4iO7V8VbdfDdKpA6VjoLBBrvoxLMf6p9UA1IjOnYH8py+C890SG0
oyFKaGVibYBYaB9nexbCvKzd3L4hq26UxDNiE/bd5Lb8bwd1a8tVapNfjs1jLaKOOvRwfaI2
hT558ammPvdyksD0vicZXu3HrPGBHMA4rG3Hn44abcgF69JvnJaodVFzzQYPN5EpfKHKGh0E
EVhtQ35mfmMn9h35frUqEfNYxB7wZ823tud7U/1BZ/gEhym1jlxicTgtE/zssbpUZ1m1t1mC
5ZqCJP1ws0fbtLQZNLVKKYPc1fJdG9KD0EruA9Tg7VNF6xn+eAdhoJ7h2r5Qup8fAAWZvr+c
wXoKveA7zdq3uDlJMlfIPnaLNNwIPxLkwqtPnXm5+kA7Ax6AYkyAfVSd6I2kDMWSM12pyZ6D
GZFEsBhERng7vfsYqyjdP8R415QWh7mHT7sGpw9VjCTNTgj4BidQ5wWBZ4I33ARqAe6NjFXr
cTGStUIYzMcayCUNpTr+qO3XMGDSjjvgJMyQYSOjKMvNN7wvm5Q3b8HtueZoPSfVK12KGkB6
yWiZXJPD3IyC2d3qILwdH7dBAYBI1TVsTHwq/rw8Tt+2PMW+5w47xpR1/biDZcgGhxtxOj0l
YYmWztrCx6s5Rea1GCFj7tKmtGKGrtUq6Omp01oERBrew5H7WXt1ibhGIXcYespWc8okCL2T
jYTuacJle3gn+B6bUuThA9WYsi/7qvLrq6yxD1NFZS0bIRP3vrrlJ71PuVvuVQOUdva7AlqG
UbwBQrmKEa1TYKJSobk7El3Agd42XRl/BYoh/Imtc0pMfzgd4eQgxTzeUrDqx5HD49HL5faI
Xs+osfHamnWYvbmQve1DtFXaM7Eys8iyBuPljpJpoQQIbq0NAZvfLGMJ8YWAROMA5mtGQ2Z0
OPsmcKlPF3LxUxakZfULqtmblGwqztrw0753HBrxqjhk008OsDyLMygMJ8f98bfQOiguYLgL
0zCY5GKuCIxMs+w/SLFPAEfPN9e1RhVKTBoGgQ12Fnu12zN0/9FLxtHhQebaPsFLc9mKxwzV
hO/cJzUoK/RJopPl3vxtxsJzv3CZxchX2NgR6UJ6faQRi85ksW5dDt3yl9eqfl24NXbnbd1Y
1THDQVFX83wb2SDk5/lLKlDRhUKJxOnHJ8KhI+Zcy/TmtbW9Kwj1S7gfe5ZtPq4DlwCMRexB
biBiQ5HEEbPQQ6QSBLVXC3jnCfLykYUE7nAYV+S4vcMawNt/tmgd6ozoPoo27Qj+Ae7kqOkf
HSROZYDDQA7a5ja1s8ll7E0AZ5bTJOeveSZmfrYho1WdnT6AcTMQpWH+HSoyTkIXIfus98+E
15kZNTR3f0w+yXg+s4fg3j+Qrtnrg3TUkQBLIT9jZ8GWi+laso2nV7qXv/IXRqVppBbfe+y1
ULPDPgftpdwHMOQSAkbsthsmw3PWHAtTNaPbjDYDGO85KVEZli3sHifMxTIsxXBSChpVHJQb
H3C9iWOqDOKynrHeYsNONs0gTGFr0p59cA2PYkF7+Ms2l/eC8mTOSyG+Hn1EHGchIWPI6/QQ
7UXzmjTSPFQ6o54uSN5eWwUHhlc8iwGeGN4hwrDD9fWYrGcUKH9DCsRipZ6H3hSnYrOHasHF
7iZNKuEJWtY2KP/fzfiYyXAxSz9GQgMqyDFJuyA6z25s5ogEsqawU+wbcCRvZNlfWimN9SB+
SktfTlNNE+rl8ykIRz6jewP2n66VEFJ2Wwy2Hm7vHejC6GmF1pX5Z8UYU6LKQbKDmEZZI+7t
Jr7XIR3TKqynkfyyWxyfPHpJANK0PawDNM4d3TOFK3ddCGpss3mKreIMrEmAzwzjoFClj4u+
NWxwvqE9IeUGKVPT6AIiUStejOp9Xd4LejjACDjEqd3SLnoutEQZdeok3mSU0siEOad1j+y9
poOFjGlYuQ6QKPLNTL3yIOyDtCsCBtWxKhP5nY90fSLNefZLXNSrwNxxIt4AqpTAerNyp2Qm
5m4FM2A00TPjayIFUS12ZlLH7jX7GmiN88/WyYMLQmXnnTQEMBmnj7bNTIOjf+XTOhptuFgp
EeQjYQhVlHoCDvrG+r5pCemHm0RjHCYY7++TsaoaXCWvidfeKRh3a/p/6EHZhiO9WU8cKq6b
ZnG4zwIdRTUTIwoPkY/mcX4cfXV0RaCai/VV06XM1wQ/uHFzS3HhFlCj7+7ICaqQL3glYeYq
Rj3Lzdy9rNXc1LebWiY/ZF//I69mrSyNqMO0JlQk+UCmtNxgS2tIWAXSEFqWv1Dy2iFmBhdg
tyUjh2b+FOUjRAZVsMRqjpB9TbhGJaCHQ1Cu8bNWZQBkjouj3II0/Bw4+8iImHPCJzyfP3B6
5BprNOWTQe5u7vL8H4b0wJznX06d0tyj1Kr+3On4EGcKnbJrYz+XYUn7HXDA/cFtTv7yKGHo
Vp+Lc/J1csiB0DIrnpjD1BfcIXuVwKIISw6HluVOwNvhw6bomLi0xzvqCYwuQ/ExFRiMFhqr
a481aLdP77mFOWfBlANUw4ikSLIqjnb/8UlAtvpTgfyCqFHC6ywP8zYbPM8HXd+NCtiv7qFg
IIHDzAHBgUrzg8DBAOCBwIACis020pTnVJ7JfkIgcdCJEXuQSQGD2l+eqj7pCgCoMlaSxHls
```

oqobCw4q/MQl4emYMiAKISM6gmJetrdqLCos6fLImYmlzJPJLeDfgu4KxrsJ02HCl4Sgq7lE
b++tThDa6Z/KAXT5mjfHwnPk2sGmYtJuSMITNAemd2k+gnxXLiqdbQZr4EhpTTsqqi8/lAi2
d2gyGQ0SXq8slTZv3JRymSLIAtDRqM27OTPFUiDs8SJM49ARNPOy2CKSlZrP4PPY4qk1Ais5
L2Oqh5BbFIsQTudCuta5o6YvbH/BU4v7HvKOVlpIpZRSiC4cZe5nnRRUHKv/wZoQgjuwognn
pI6FcCBfIWiF4kT+7Yc2U+SAJBXleEElctaNlWJRLInBluBem4qsUDz2VxapptzX44texlRR
7axphOqL0miqQJUKJ2WdvSzPIO9baF7+koWozd5xfapmXxh4eZqxwfdjf44MprjYocpD/pGU
YNaSs+n0leqoAcke96d0dUOBVp8pnWvzVYKwKHjLkEqU5CDxFHozb8SvUtRxIFhKAF7cuQaN
Zp7HIxMwUYQ21+I+6pW7m+06h6l0h0hInLU9NyCFz1guZHY4tx2tvrA89PbQQAgGzYqU7XO0
RgsSsjtD3UvGZD6U15DDVAEUsfFfpYq92I9mmW8WUdG6MquuYR4tvsraahggzZoEmdgENPgn
UKA0g+hYswUqsavMnRINsXMSrVCAX4iMMD4Cj2kyepie41kebqMhLZtWrCK0mixHWUsx/UPP
LTI6IiAVbKdWUmUTAGpaiFkSa6RwymZ3ojBh0ZGqQwXvcWLwAe+XhRuKRAPWjBAgxqn1Sdkt
3lLqWOHOcPErpQbN2uj/H4VqNahLR9YnYHgQ2ucto5gaRcCn5vu8rUWjWGgIng4MsA4tbL4o
MGRubsC4S3m6ZYTmA89MM4Ztm0MSzm9ZTvphUj8FxmOmTEr6XmazkLLod0woXJZc2kZpJunF
3l3D8VOpnhSBkMoaR26QBBt9gv4zRnAJW6t0j/rzZM5kx4S5apcRBpIw2GmXg+dsbWzlSIct
WysFAqKatYNE5Hg0YRj0c44KggIvJhKOmxQhvTHzQ6YPJlDnc0eInI20+2Xb42C1JCPuYvSf
muREFblgJYevDIgEw6di/Lc9ZSF7wkUTyDN2ERsmd1LdRj1q92VkpGmFoERPPXdDv9cGKAhi
iblqtawkvrYSdJXz5sZdmYbqFhsVcpniiqIWNzp2OiiWY069IoEr8l1Qq62Qc65H+gpqskG9
KuItaYZ+3q3Ixym5AnhQ+T0h1I1A2MJgRKyyVGCCSBnpGiKbR2mcR0zAdlVTLrETRKThmKw5
3l3mV9EiZqkZJL49t/ChYQe6ahLzLNqsf0oLJ+FOr2A+4dO9yCm8aCErEOmLMF50cxpLk42W
Cx6VjGtFBXc+XI6qleGVXX3iG9KHyBK+/mluyHYCHNmFI4qCWVHid90N6JwGg0Xf2QBzNI/A
RtzwUZ0Nt3YAU0UO8b2gObgkYo0T6jyKTMuZSiAK9KNOnQJ1xUUCyZCtDo7kLJLyAvm0dKAc
BTFrmmlaPxvj1eYhOT0bbxfXN/Dn1C2F0pqJw5uMK9wGZg5E2samnM2xGTuJe1dqy/IY40Ry
K5ZMZQ+DXooWO1m2u9JsNkKUcFGqCji+XqoCihJAXU6k1v/nAHMGxXDix1M/AbbFnpKvDVJU
NUIV2j2T+r6cDObnR1GB1f5Q6gcOX8niatmnnLicXBN4jCdrRh6x9mPJuBBwSgsguRHhVDL0
gXYQkBtHGfJGxQgGFFmboEiFXpayQXkaZfpmRfaI5ajindqU1PljkyVBqgr12bo4lriW7IDR
hY38bbUDRZ+rMSGAntqdIIoiVTXThCegPJrT6aol/EB6thiYVPoW9ny6Gu6HoVDVBQLNgK10
qg66a/l2U9chpvZZMBMfAEWw+QRLvjrFphFwm3epRLPUZSQOZ30/xnnArZ42bvPrxHdvDQvK
dgWEMVgjftqspD5h9he4TYgyZCQrLLXg4og0BLlM4XpIxnnlqGb2yOOjLATAVelsIk7ik7th
68+faC64dCct3waMmZDy/QhbSogKAchXKGvxREAAptxZuD5x0Fu+RZD3KzPPzq5NbdsWmqwE
46EkwZN5jMevFMI22fTMEbegBAGWkN4NfXCrSGAjhFLQ9vPejQvtCRWpSDVqDfhgXqZjscL6
k6CzhgutWbH8QAmNLvQyIb14Y5EnbYhWcD/tR4We/uPerllsiEyAHZq+tdARkkAn03JVE0Bx
GOU1gCuPuJpLhgwgZswEAYHKoZIzj0CAQYFK4EEACMDgYYABACe7fcRLUKh2ESJPrFIHMHXI
7dmWI1wcsU8uKT0v93+Vv0YVE+LD91AsAww9UZo9w5eFh6vDouzG8+DoxVen/gCuCcHQX8To
2GnQ70El/sQmsjAt+Q7B39US1fxDcD+Ek8GGtTakgR/hMf5EklevMPw4tdKUQtcRD6PiyWNU
==
-----END PUBLIC KEY-----

which decodes as:

```
algorithm: AlgorithmIdentifier{id-Dilithium5-Falcon1024-ECDSA-P521}

subjectPublicKey: CompositePublicKey {
  SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: id-dilithium5_aes
      }
    subjectPublicKey: <dilithium key octet string>
    },
    SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: id-falcon1024
      }
    subjectPublicKey: <falcon octet string>
    },
    SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: ecPublicKey
      parameters: secp521r1
      }
    subjectPublicKey: <ec octet string>
    }
  }
```

The corresponding explicit private key is as follows. Note that the PQ key comes from OpenQuantumSafe-openssl and is in the {privatekey || publickey} concatenated format. This may cause interoperability issues with some clients, and also makes the private keys appear larger than they would be if generated by a non-openssl client.

```
-----BEGIN PRIVATE KEY-----
MIIt0wIBADAMBgpghkgBhvprUAUFBIItvjCCLbowgh06AgEAMA0GCysGAQQBAoILCwgHBIId
CHSCTZxmMzvNno7MqyymsTr0fdZNpggLUxyORSAPe69I5FjzKTYFpPVhB7JGUKAqSznDCS/K
MW0zqC8zFogMoPxhy9c/Vx5RyOr1xJ6nHvu6aBxQKrN9G5d8hABzPepIZCAhDAlmgYkwHjMk
IZkNglMAAGAwI1buEgEJzBJQG4awwmQxJCbOARJwIEYKWTDEHCSRFHjkoTStHCEBEniQIwjF
pSyZEiQLgWEiMREQlxBcFgDbFGHhFk1YJFGAgmxUQoDRNGHAsFEBBWgAFmBKBiwDAYDaJDDU
YwkUDuU3jAmxJQonhCDEMFTKUQjEUESAAoIzjFGEIR3JcQo5ZBpKDBiILGQIhOWaJliEDRCw
xcMCFkElCSFGKcslALApAYKYpYxo3ZCHGbMigJBiLUwo0TEGDaBgwSFYKkNnAIJQocx20RBA
kAiADGaJmYLSGniwBEYFGzEEnIigYhbSAhKMhEQpnEZAW1iGHJQsmRTIogMoSkEFFLTgCQEG
AITZQEzYAA4TkQwCmExUSCWSBGBDkg1MBIGRJGCZpiHJKGgQJYxIIAgiBWSDxo1AEo4bQyIK
IJSnZhoTDBEWJAilUooDQqIBBkgiUgDACo0EUJyKKwmAMSQrIFkohMSnDMgBRhJBgNIiZMC5
qESFIjBIrDkkzCyC0ZKUWiGC0RkSligAFYwhAjAyoKw3EJqSwAgAgBgDEhNRAQSQLQQEFCMD
UTiECASsm0QRSxbQJLBtgBgFjLbsIkjQBBMQFJMRE2gtHAEqY3aJo4ESJABJmkDOSFbxgAER
MlLZMGWIpASjtCwDCXEDA5LiSCUkSGbioBHAEBFTOAYBBSHTQE0MBo2Tgm2kqGlAGDFQgpGT
gQhEcQRCAFmYCpHEiFGWDEmUINTLLCASABI2Dwm1BEiAAkwFZFGRhJiEToiSEsCRCNmZSJJA
kQBw0IokBYskFEJgiiIAIKBCmSAGbAMhFKBgARtA2AMGCJMo6DFkhhCE3LqI3DBiaUponSJD
oXEKHEImECURmDBSAIKQU5KSEJjAi0CsgRRBpJLKG2AJA1JJobhJALSwAXKoCAEADEEoYygE
loiglgyZsghDRkrkhkCKkGDIKA4aEI6hRJIRNE4cEGTAECajqEkRIyQMgAURp2UMBUnEKCBk
gAEgQIiEiuSGCwAjigoBQRAwSRk2EEAQREFJjMIkIEy4AFwBhtmwkpYQKoBGQgA3choVcJCA
0hNyRAQC1AOI4EBIbAIiVMEEVjQhAiFUBLFCaKBC5DNmiCkiwaBAkaRCDEIJASRE2YRlKAko
EmMmIwMs22LOFHkJG0aQSlcuE0LpEwZtXDTxkmRCIWKNAQcRCEABFEkl2EkQ4AICWngGAwDw
RookIQEYhzGMFIHDpIAUlWhSRoJZEjEBlBAgFC6buImZAgHhRooBAXBjwk2Mlo0boWEApYRZ
csimbMAkixC3jMhIaEwmQlhHcIi1kgmUcGChMgkyJNAzYhokkQ26hIEqQRGqBuAEAkSgEpDF
qUBI5AqGxCJmXYFoRakgwaJ00ghkiUpI0hhG1ZmHEgAUwZAmGJNArEIk5LCDIZBkIIkUjIBg
DIbQUHaGIBRCAAah2QUMogbyIyCJmkBQixSKGHcCAZCoizBtAjMwmkJwgkkCBAEuGxhQmYAw
k2GgRBKZIExJoiRZMmaixGkRlIUcmHCQJIWZOArKSCbcyEUYICmEOCDjQAKYNjLSOJECNoKQ
RAi1RSAHhhIDgME6cQkgZQQEIKEVZBoIASVGYxAGCtm1LJgwABUjCxHBBNiUBuGDilEwhGIA
HTAlAMGCggqA3MQEhZgomTMnHUOJIKEwSZok1YxGACNQ2cJGmIKCkEwTBUgGHKwIkgIAoAB4
QFiRghJKHAAGYHSRGYCJwkRSDBIJm2REiAEFYIYoyjKpCTLkEFAuEkLwACKJITQBHBKAFIRB
FZAIwWWSoECBECLgQGjiEgkJREjoGjomhOi1h8Z8ICorSdRFQp4Nmw2sHJ+4+oc3CjikdZLN
RBNQ7N5Rb2dMIsCImgQB++SDZQ+nl3hJldUjxQAFzKyLSb5F+4hXvLRv6RgOzBUK/CquggEu
gJT1iAjLZAngXa75ugw2FCY6B17b5hy4SNCYx1yVT1MuLcgDDULUZaN30EaSs8CceG2Upsnv
StQnXRkGkY9y5dLslqqAQA6/g9HEODmgiShn7FQkK+mxNSKSbxcLC3tcf/3yJAEPVkbiA1TE
QBhLeCkORzn9dEIKH21yv2bnZcu6xGMTkg6mwCLdJ78fxZYqc4W5hpLsU/Ky9jWLJ1a9UAHP
pnr1/5EqPwo/L9uUSwsM82RgrqlQAZQaFsYWt3jFWtrvIKR+ZV8j48NKQf2j1vL3vbR3RF6T
tyGu7R2X+kQT+Hyau78eH8bxyWenaGVrQaeBlpjzx4iyVwGE/qvikvlJoSIWm6Uxygyzzoqj
1chO3hMr5OLIiM0LOkLrd/A5GFhufo9x68NzD3anuUil1vQuWD+obD7T4MxSZzHfRsaqFoh3
dOaCex+ZfOkX62qRFM2xOswxKYPpvZikzDNZ61WfQFCdYKeGVa809MgdcqCeenKb+FzecPW9
P7GCA6knVcKRZenfc+L29sAFoB6+wl79mt/dQX/81wDMgV3O1imdPuZUqq9FS3XH3tVYjpZS
f6y0fi7VAhil5ZwFfYQa+9V3ALvRVo31UgMwEq6lEreYlPHXzfaJ7rrToNi8cCEuS99wfEkO
uK1VrsXl+CulpoidZ9uX1HsxsSIejZaxfkJowcwAt+51o6fUxBFDMo1FDE8mvh7pG8V28jN1
qe54J/Mm3zyqnfNy1OXHWYWJWTnjBTOOgrVbHQ9AAazq8U4nI6m6PD8I7xfukh0eJ6W8xpWX
3F9kwe0gXF7cGIuHLTy5uCNWDn2ANjTLqgxzLsL/lQ/NFEOoKT4DVF6rWXqYdQp/Q4EEHkQO
kw9zzIK7aMhSm5O6FPQzQwbLRVVZLqp8Z/zBJDKN9s/MtI4BQ7FHfCVuPfLVUJPJ5baFtLkB
G+7s/eOOgpG9zb8Nk6kRCNIYNOtuNcnkj4JfwDPCDQL7Oagl0ecoRhVo42yWk952hC/+TUDO
Z941MQ7aRQ4XvlK1fgClYBEIfg7Zg5RyZg/xt9eNNPRmYs27QTDScba6cT7dGlEVkDox+364
QznxcEAepPedqt2rYFH7gqEFS8wW5LQMEqSkD/CyQJhGlmZIQT1IvwC7Q+iRKfIiLzyG30j8
HVRn05VvST60BVcrDXgLgdY49STnebgJJES6cW2b0/hyNJh0/QeVWQE5xJoh6+D7JNj5p51v
4gsEI0KppuThC+HJ/nWyerpB0PQh184dkIn/SdwAR2sJGqznDLUfYv7ynU+bgYwId9AKdEvp
```

```
SFTtVuToHHeYr+qHRDD1Db2wi0TwuoMlQNZTeAlisRwP2bWLZ4M4OSIjPaJp3TdOwxTAxGDa
W3q9l+RGRO75XpBXtlkGO/JXkTvJWoucKxC4JBlfiGKjMESes2288fUHRQHmJANimxnvs7YI
5Jug8y3Xt1MU8s1mxB0V1eUASDDGCG4unzahp6sHswFNkcOEa7ulP5m80wvaXXRsf3s8u9cz
fCa0uHYmEA+W8VxXPs/chuWzcTpG1PG3bfZeqNvg5ZeV579F2c+zCePjGBKVf0iiJKb1YCL/
zix3jxxO8ZKIsVsXx+40AhJoCuJzfzugoN1qvWu0bqw+nMa88JyqPU9d4sKiUQzCy9rsOggc
Y3kfMddkfYnICNUB4Yko14cOxzzOLj+N7B3Viir+A+BtOf13++Pl/JRk2kgWnruu0l5o8cwZ
CPyFVptidXxzsQvwOqOluflUl8RLmcxFZWuxnmODAzVtnSXSCapwK28/D56/FNGm7OVlPeQp
cNKqIPJppiTmQtWR10675uczYMT68X0bA3MHZTQinCaPA+f5CgX05e/ypERDi3eAxZ/XUQzv
e8kNsHVDBsrIEmK8bUf6nYCpED4hdQffxGulZLJ5z3iwubDYqqH1OPppHT+0eGdu87/FfZKi
UTFsvH+HSqSDjZmfTua1eKMMoDadSiO8jGoudHwxOEwCg3Ku/CJp8NeohUQuEWcWZa8Ao+xQ
JSu9mKCF6hzyBP26AalqskNRfRTyqoOijWuk152IX/Ex2ew2tHbOICcb9gzoBfVWlwNeeUFy
G1a5p+T1c8bfjSTYDELnCjmflndAyl9MGzWD4xxVtRgiM4Jr53FKyWBRnBxX2WMDpaJ4D8ew
FN8vvGAa4XzejwJS/y6ckuLSFMxdtBXd7L1OUiyuGZyXAAPryq9EdQCTtFm83Ypvk+97iPoU
5vWKY8Z/j4KFOiPu/CIRCrGK6O+/YGhj4zZOuxcu3pTlkeBFrX41LhCPlTMqBlt0KUjUHUcj
AE1bxxwlk2Z+U/bs0PcmTq2unL19pO7n72JV3QKeSGK9NF8j9wN1l6iAJnpq/28v+tHXhoEW
nsMzDg05P0FsIVIF9W0mFJ9KsmgZVTEvYXnw6Fib2wvh49atk4a0nF58FDoKO9gux9JTKMhP
E+c5nGREFxw9VuLViumLLdsKFgjI0t3/ATe0b9VnLRdzhKzV8d+eLiAeuMnrm4g/Fln1Ezac
OhHoFQx4qX/FKps6DE0782OEUnAQKl7ijfZ/OGTLpKLnO5Pca81GczPJVpu2GghJgRhCYU/v
/zruaJ3Jzngi4Jq+tfo4Z4hyPKSij63LGNtozrxVjp/G38rmhKrkAK0k6be3QLFjJWshvB11
zOmfW0PeyN6FvbG4jY4fMPiHYTikBaFccQEuI9qWwGvYv8qoqCsV1SfeKUq50NPbypg4Cyzh
OUmya3Zl7+lFk3c7021dB052bQJ6gIA9Ulru065I/lApAFesYDgmKu9W/kJk71woNltWedqJ
uGSBKt1BDW6dG1nZOzrRy2hkvH+rGL7sJzgbDAx1/qfZn5axxPXYXJd1rNuIPR6Wc/ixCHeD
hblq9AYEeYQL9oBZysdMBxsrReDNUIK+ZItzCIYg1cPNPLquImO+IOm2he91L9uWZUvGXLkw
Kmt3VfFTLeM65byXOqrsslOip0WFg4K5W3MNxiNbm7cnIL/Nhw5Jk/IfLnzKqUEGqJUybtGT
97iFNIVR3aHX8R2l+bacQ88dwNGSJsW7K82hNXO4FlXp2JvkAmOuRssycB9j2z/6S3Vzgryz
wgHN10EJ5TJb/KXCZtjGMKNj7EBN7o879dtyYc+16Fc64SEMpKJIc1rcPoAH52x0pT+nZDUZ
eIA4h4W2QP1HxXxgOnD/PbaIBQKA7j3o+TTVCxgrOyaNmpgPqMoCp+jGNwPpJ4S5etWd6IdL
4Nt1QA9sdvXWwS1OU0avGGRc/4ROFhIYCdH7H7LOXqZXnoGCG3JLepgXJ8QxrUf8pZ9rdaBM
eLzOw1kiX0S0wrowWUlapmcjspYHjfB/1x5Ob9SlA9uodyBvRVYCBsJvZu4OBIELuPeeL4g4
tqjK0o5iV0QPzNBt8BZ4FA9PrUb66o4zKAnIo3u3m/K9I/AY3G+h7nk5WW8tQHLLdYg/NRwe
XcRa6yVix6fJ5qJDT+010FOk7gCmnevpoqHRGWU2+h9CviQdRtlAiwZLqA+dXi/DAVBlhBsR
SQjknZiOnpHE9aEY+BGIddFzQdZjFbXxmJZEhMolGX133AaY/gbLVrumDNPuEpPeXw1bo/cO
A9wM7MUN02QnMDIfGFxJFIrORNI+CwGssrI1E0ad2p1Of7WPGBawCPB/tfonKveNO0og3utF
LBOQcJQIstjZMW2IrfJ5cMzBC3bAikEaBSmvSiUgMMJbv2A5p5y15TFu4X7S+/2L1CALutcc
TLIJetH93DLW9Ewnzyhd6vtQDMr66wxtgqwlGf81FHCASMpgJ++zF+zf7XJav48y9XAHZNPa
F9ly8kh4TvCflH+OP3MVGOfyfwMtUgcPNqT+TCrSyWkZk723HZnI0wz4r561GbH1UBuY9blE
LPKesLxMgz9Xhkj2ItOwrIRxxZ+MIzOVRGKw+iOnbJrAqGuwfw6pv4kw3MMuIhayaJXK8w1o
Wl4XR6bRPIsrQ1x0Xc2pU/PcRXTdJ129qIKXCZpqpYzbc1rCXRhDS0CGMa6kN/8+tzmymDZ0
I0m9O2OnHqfxQ/H4oqV+YdF+QH9ymvs8XjNlkoa4NwHV+zU5sEYs+IGFPyJU8g8qsqQ0n58C
v45o1fdmGAw40xn8Esyy/uDgLBZhH3KH2VnVk6Rk2cZjM7zZ6OzKssprE69H3WTaYIC1Mcjk
3uvSORYQwbUPEZFDach1iOJKkiXpb2nl91g0CJJzF2euQat5WK4jeS+vE+0iAGyRERTgO1V+
dbL/oXaltcsw2e9eaVkpXxeq7r13LHc2myoarAbUPhZWw70okF432s0RLmfacYE/uwj3TlrI
7ELG7r75Vh5dLx+m4mDDXSpmdBe7z9jb7s3UIsxvsLpPm8ddj8QOMY+Jq8YMvEKYWoNh/4+I
KQo1jFOuArFYBRmEZPp+CZDjzVlvEQQqDOCIriRRwqRqoM0oRtpfws2gHHZy8VOqHZAtGdNh
vGo6xfcWAdFYxQIcNKHzZTlPh0MZcCrzd25nG0hgUe7dJkkHV526G5eYLLThmUY6b2pLH+LQ
szrUHAl5pch5KYgQ0S55Ya5cTotRvet+Ej7URTrpERuVSE5QftfB9lt/mcWzTQMEZyhH8MvT
ayNbo1DbmDDLYnX7ts2sG+tC64hWNsMQQZ9oqXA1++VLC5RES4NxJeXDRRCUE7rR37XD6WJW
nzEix46lSIT2YtRwVqKsEIzpjHs1BuwoMOAsVa15iLi/rx7Cxw2r4RcAH9NviaKhS2elGu4c
pJhD4+bMi6pJ/jVU849nhfpfogXLPB/2HbHKdd3+6fs1nnBfiPrIu004g9QPNdKMavtUA/l2
```

```
w7Qz5cgry6rpK47s2zicHYvEjHtIKCu72/9stEU5qYQ/7vj2cfLzJ8WcpKFNnSZu1O4iO7V8
fDdKpA6VjoLBBrvoxLMf6p9UA1IjOnYH8py+C890SG0RzmoyFKaGVibYBYaB9nexbCvKzd3L
26UxDNiE/bd5Lb8bwd1a8tVapNfjs1jLaKOOvRwfaI2ETdhT558ammPvdyksD0vicZXu3HrP
MA4rG3Hn44abcgF69JvnJaodVFzzQYPN5EpfKHKGh0E0w1EVhtQ35mfmMn9h35frUqEfNYxB
823tud7U/1BZ/gEhym1jlxicTgtE/zssbpUZ1m1t1mCL885ZqCJP1ws0fbtLQZNLVKKYPc1f
9KD0EruA9Tg7VNF6xn+eAdhoJ7h2r5Qup8fAAWZvr+c8iXwXoKveA7zdq3uDlJMlfIPnaLNN
xLkwqtPnXm5+kA7Ax6AYkyAfVSd6I2kDMWSM12pyZ6DikZGZFEsBhERng7vfsYqyjdP8R415
7mHT7sGpw9VjCTNTgj4BidQ5wWBZ4I33ARqAe6NjFXr7FfcTGStUIYzMcayCUNpTr+qO3XMG
jvgJMyQYSOjKMvNN7wvm5Q3b8HtueZoPSfVK12KGkB6s/oyWiZXJPD3IyC2d3qILwdH7dBAY
TVsTHwq/rw8Tt+2PMW+5w47xpR1/biDZcgGhxtxOj0l9q7YYmWztrCx6s5Rea1GCFj7tKmtG
tUq6Omp01oERBrew5H7WXt1ibhGIXcYespWc8okCL2TFqcjYTuacJle3gn+B6bUuThA9WYsi
vLrq6yxD1NFZS0bIRP3vrrlJ71PuVvuVQOUdva7AlqGZH9UbwBQrmKEa1TYKJSobk7El3Agd
Rl/BYoh/Imtc0pMfzgd4eQgxTzeUrDqx5HD49HL5faIaUvXs+osfHamnWYvbmQve1DtFXaM7
8iyBuPljpJpoQQIbq0NAZvfLGMJ8YWAROMA5mtGQ2Z0G/eOPsmcKlPF3LxUxakZfULqtmblG
trw0753HBrxqjhk008OsDyLMygMJ8f98bfQOiguYLgLInc0zCY5GKuCIxMs+w/SLFPAEfPN9
hVKTBoGgQ12Fnu12zN0/9FLxtHhQebaPsFLc9mKxwzV4ULhO/cJzUoK/RJopPl3vxtxsJzv3
chX2NgR6UJ6faQRi85ksW5dDt3yl9eqfl24NXbnbd1Yj5W1THDQVFX83wb2SDk5/lLKlDRhU
OnHJ8KhI+Zcy/TmtbW9Kwj1S7gfe5ZtPq4DlwCMRexBXyJbiBiQ5HEEbPQQ6QSBLVXC3jnCf
YUE7nAYV+S4vcMawNt/tmgd6ozoPoo27Qj+Ae7kqOkfq2jHSROZYDDQA7a5ja1s8ll7E0AZ5
OeveSZmfrYho1WdnT6AcTMQpWH+HSoyTkIXIfus98+Etud15kZNTR3f0w+yXg+s4fg3j+Qrt
3TUkQBLIT9jZ8GWi+laso2nV7qXv/IXRqVppBbfe+y11wrULPDPgftpdwHMOQSAkbsthsmw3
AtTNaPbjDYDGO85KVEZli3sHifMxTIsxXBSChpVHJQb1CkH3C9iWOqDOKynrHeYsNONs0gTG
p59cA2PYkF7+Ms2l/eC8mTOSyG+Hn1EHGchIWPI6/QQsUT7UXzmjTSPFQ6o54uSN5eWwUHhl
wGeGN4hwrDD9fWYrGcUKH9DCsRipZ6H3hSnYrOHasHFcu77iZNKuEJWtY2KP/fzfiYyXAxSz
gMqyDFJuyA6z25s5ogEsqawU+wbcCRvZNlfWimN9SB+sCkSktfTlNNE+rl8ykIRz6jewP2n6
FJ2Wwy2Hm7vHejC6GmF1pX5Z8UYU6LKQbKDmEZZI+7tkdvJr7XIR3TKqynkfyyWxyfPHpJAN
awDNM4d3TOFK3ddCGpss3mKreIMrEmAzwzjoFClj4u+KuUNWxwvqE9IeUGKVPT6AIiUStejO
d4LejjACDjEqd3SLnoutEQZdeok3mSU0siEOad1j+y9js7poOFjGlYuQ6QKPLNTL3yIOyDtC
tWxKhP5nY90fSLNefZLXNSrwNxxIt4AqpTAerNyp2Qmh+M5m4FM2A00TPjayIFUS12ZlLH7j
miN88/WyYMLQmXnnTQEMBmnj7bNTIOjf+XTOhptuFgpDX6EeQjYQhVlHoCDvrG+r5pCemHm0
CYY7++TsaoaXCWvidfeKRh3a/p/6EHZhiO9WU8cKq6bfGjZnG4zwIdRTUTIwoPkY/mcX4cfX
aCai/VV06XM1wQ/uHFzS3HhFlCj7+7ICaqQL3glYeYqejnRj3Lzdy9rNXc1LebWiY/ZF//I6
SyNqMO0JlQk+UCmtNxgS2tIWAXSEFqWv1Dy2iFmBhdgwVBtyUjh2b+FOUjRAZVsMRqjpB9Tb
aCHQ1Cu8bNWZQBkjouj3II0/Bw4+8iImHPCJzyfP3B6UDe5BprNOWTQe5u7vL8H4b0wJznX0
tyj1Kr+3On4EGcKnbJrYz+XYUn7HXDA/cFtTv7yKGHoiw6Vp+Lc/J1csiB0DIrnpjD1BfcIX
KIISw6HluVOwNvhw6bomLi0xzvqCYwuQ/ExFRiMFhqroT6a481aLdP77mFOWfBlANUw4ikSL
nb/8UlAtvpTgfyCqFHC6ywP8zYbPM8HXd+NCtiv7qFgxbMIIQFgIBADAHBgUrzg8DBASCEAY
ACWv8CMwB5Dz2RgBwIufCPvvhILYQBB8HfgAIXRi0PYtgAD4NB4AnOhB8KwdELmQg94nAB7/
GAHvf4MAP+/0QA5B7pNA5/ROeGMIQgB8QA/B8Pv/H0IPdADnjC6AHveAH4fh9/XfjEAQO9+P
4EAPBMLvhm+IOwAB74QECD4Rn6AvQeAAIgiEUn+8AEAQcD4XfADnvec94g//GIBO7GAJRhLn
6Hww958wgB8LnPfHz5fBEA/wA6AIOiCIAvjGL4PfL44Q9+QQ9+98XRB//3AB8IIfmEAWwkCM
D4hfA+QaPAGH//DF0IfcAMHQhBsARBEIBA/CD4AlyEYvhIDYxe/8fh/6IN8/yAAQd30X/h8Q
CEYwfFrXAB2b6Bd6QhOEH4nwDGP4hA4PQR/HoIDDAH4QD74P//B0Aej3/oOAIIROD6P5dEN/
EP+gAF//wgDzowc+EnQ++L/wDD73gfIIAjdD8Xv8D4YCi38Axd4P+hl0a//CCIAv/B0OzdAA
ATpgiF8HBh+EAPBF/3Ok94IfBCAAPDH/3/g1wABg0XQCi6MIQ/+IIee/s4Q/ET/SgEIHeA6I
1vwvhKEwwB/4Y9cF/3hDD74PhELpxAED5Oh2MwAACEgv938Xkd97Yig+QoOiBoIB+EMAAf+M
/0P//7vwAAEQYR9ELgie58vxi90Gwb6D4Pj5/oRfB4PyCCEAPB+QIBj/8PSf4L/g/H7ZA/6M
93wy89//QA+Uwgf94uzeKIXw94EJP7+Puwd9/gA/J8YABEH/hC6L/eCCHnQ7/7ovg8DufECE
F/ZABJr4gEGAZSgDoPxD6QQREB8QgA2b4BdGIP+/8Ln9j58gA6GMIfCKP3gfH4fRAH/fvjCE
```

```
7sZOe6UReiyYAwhEEfwj+IBObCL3wd+QoQ/5z6RfIDwSf98YfF/8IOhEUHtAKToQ954IQh78
F0RQiyT4fgAAWjE97YS8CQfxAJ4AgbAUIAiD4RycD0Xj+ALwg9+H3dhIH//CB7wAi74IBi8A
zz4fi7wIfaB8gfe8EIiD9kPcg/4Xh9JsHBCCQpO/78gegH/u//D4PAc+T/fG8H4ggB8YQC73
CIQgf7oAOD+MfPfF8Hge+AfRC9/4PD8HwO/EIHv85/3fcB7gO/B0XP/+IRPiGEIBA93Qh/AT
AX3Qc8Lv+kF/+/g8EPBB2MQPfAL//e8QQBiAEBP9GL5jfIIIwi8IwP7zsYfbEIgAiAT/gAAP
AAYidD/hPh8IYeiD0QBBL7ohB/0IA+N4IB8575BA4AQy+/4fwcD4PAcz8f+fB8Ac/CLgQ7B0
6AYxkGLwgCJ4ovhGDfP8EU3vhB8ZQc8AOh+8L4PgAIPfGCD4+b9/4SgF7mv6DwQP9/zmv+H/
7zwAhCEIfm0MQ/H0Lm8iB4XijL8P9+D0wgl/zoPkCEIwhCEPu7B3ovB/7/9hGLwSbD7oBEGL
+EIihD4BAAMAXw9/zYwfD8XBGAAYhE58QwBB8nvf/ogQi/rQ9i6LQhi533wFD8YUDAX3iBB4
B8XvhFoAA/EEHxh6MYvc8Q3BZCABfCB0oh8EE4PjB35RE90X+kMIYC9D0YwfMMXQh9/n99B8
CMHfBNknt+/vgNCB/og/CEYNgIDowi2L4fgyIXg/F/4AgGMIyeCHwQg9/wyA5/89CPIN1OgF
4Cw75+RAtF/L89BD3I/oaGv/uF+YwEQrrJikx8RXj7SPWCjL+AxYU3vjt6hQPAAbO9+cV9AP
bxq+wW8xf+IQAKCCv88dwjLufbChAm/f4IJfP3Ms0l+hH/AQX99SHb6fQl/sYK4QcC3u4JAC
CcPCP7hAAITCvYuBBLp3f4aCPXR8hEGFCXc6xIg3rsVBNsC8Q30ANrY4wra8AIaAOIW+R017
IBzv3g/m+PHzDP/ixwAOEy0pCfcD8/AXOhn66wn5Fhgu8gEkDxUKA+//DQb2CxHXHwv4+h0G
+KfUZ/vXF8g8cDA8NFTzI3hju/AsB/xIOUvrPGQ4RBCPqBAP3GwHo5u8K7Pri6/kf/+bHBvg
/+HO3PCPkK7A/99hwP9v4NFxvYDg/v89nqMAcpFfgWyAzlCfAYBRDjJ/nyCfYBBQog9RDl3h
PgNHgMND/QZ6BMc7SMkDQcl4xAmKPELGekgF/MR/ur1DTTxJijxEfv9FQYLERwbAx7sLNvhD
+Cgd5+/fBSIMBukF+uwlL7UkH/oO3Qz5/f747vEMGBMN//XlA9LWBBIADh/u4fMM+PDy8wD0
V5vUWAPsB7BMU2RboKiEcCOn47P4VG/zv+SLvNScBHQ8iIfUP8vsFEdoH/Qf+6BzlJfL0Ju/
wW7gEE+fntK+cfDw7tIBD71QgACRoX3Pf/Dt/09OkWKwUsJvbrG/EE//oZ8uX80NkLAtsrEe
dQJChgV6/j2ExkYDw7zD+QaC/oG8tcqAw71LCPh5+n66vsQ/u7kzuknGrzmPPf9+hwI+vEpN
Gdnk+wjzHNrrESAO6Ajj7SgL7wHuDPn+FvrtBPH15fH9GRjrIAL9D904C/QDWCIDCPzpEDAR
IAdwP6hbv/RsJHt378uQT+QTx6u4J6yMsIwT43CUjFAXm7xTwFzHfAQkLBgjxKvUQ5+EWH/E
QH4w/s9Rb58z3aKR8c8+Uy9v8S//Ef/yIh4+E9OAEJB/H0IfsBRfblDh3u5Csi/PsLDeIOBh
fP7EPj+AiQH8xUADP0YDOUX8dYOG/0KBeYECPn06DkHFQYXE8zuFuLuEPwsEfUN3+EZ/s4IC
A/Pg+fjqHAPzG9Ig/9cIDP/8HA3nAQgO/iP09soOCOkkyhId49bXGfH6I9zx+QnnDv0j8wr9
Z6gkUAevqGQoiD+4C9xwWGPb5CC3NzdoG6Oz2Dxv/FgDkJhnX5BIK6kAe5/8JBhUg2OUh2Nn
3zD/DeLAPw9MfmChIp5ybS8wL79/ArFuHo4gQo7trs7vLxBQIN9/oB5NXT5Nz5Hvvg9wLZ6N
AorNNtKU51SeyX5CIHHQiRF7kEkBg9pfnqo+6QoAqDJWksR5bJZUqKqGwsOKvzEJeHpmDIgC
OoJiXra3aiwqLOnyyJmJpcyTyS3g34LuCsa7CdNhwpeEoKu5RPlLG/vrU4Q2umfygF0+Zo3x
5NrBpmLSbkjCEzQHpndpPoJ8Vy4qnW0Ga+BIaU07KqovP5QItjMsHdoMhkNEl6vLJU2b9yUc
yALQ0ajNuzkzxVIg7PEiTOPQETTzstgikpWaz+Dz2OKpNQIrOV1hi9jqoeQWxSLEE7nQrrWu
L2x/wVOL+x7yjlZaSKWUUoguHGXuZ50UVByr/8GaEII7sKIJ54cRaSOhXAgXyFoheJE/u2HN
gCQV5XhBJXLWjZViUSyJwZbgXpuKrFA89lcWqabc1+OLXsZUUfUqO2saYTqi9JoqkCVCidln
zyDvW2he/pKFqM3ecX2qZl8YeHmascH3Y3+ODKa42KHKQ/6RlHBWGDWkrPp9JXqqAHJHvend
gVafKZ1r81WCsCh4y5BKlOQg8RR6M2/Er1LUcSBYSgBe3LkGjRG12aexyMTMFGENtfiPuqVu
OoepdIdISJy1PTcghc9YLmR2OLcdrb6wPPT20EAIBs2KlO1ztOSxkYLErI7Q91LxmQ+lNeQw
FLHxX6WKvdiPZplvFlHRujKrrmEeLb7K2moYIM2aBJnYBDT4JwU21CgNIPoWLMFKrGrzJ0SD
Eq1QgF+IjDA+Ao9pMnqYnuNZHm6jIS2bVqwitJosR1lLMf1Dz58Ui0yOiIgFWynVlJlEwBqW
EmukcMpmd6IwYdGRqkMF73Fi8AHvl4UbikQD1owQIMap9UnZLZFxN5S6ljhznDxK6UGzdro/
ajWoS0fWJ2B4ENrnLaOYGkXAp+b7vK1Fo1hoCJ4ODLAOLWy+KPdYTBkbm7AuEt5umWE5gPPT
bZtDEs5vWU76YVI/BcZjpkxK+l5ms5Cy6HdMKFyWXNpGaSbpxWOE95dw/FTqZ4UgZDKGkduk
fYL+M0ZwCVurdI/682TOZMeEuWqXEQaSMNhpl4PnbG1s5UiHLWq1VsrBQKimrWDROR4NGEY9
CoICLyYSjpsUIb0x80OmDyZQ53NHiJyNtPtl2+NgtSQj7mL0nwCC5rkRBW5YCWHrwyIBMOnY
PWUhe8JFE8gzdhEbJndS3UY9avdlZKRphaBETz13Q7/XBigIYvCbom5arWsJL62EnSV8+bGX
6hYbFXKZ4oqiFjc6djoolmNOvSKBK/JdUKutkHOuR/oKarJBvdIqCriLWmGft6tyMcpuQJ4U
IdSNQNjCYESsslRggkgZ6Roim0dpnEdMwHZVUy6xE0Sk4ZisOWPmN5d5lfRImapGSS+PbfwO
umoS8yzarH9KCyfhTq9gPuHTvcgpvGghKxDpizBedHMaS5ONlteHgselYxrRQV3PlyOqpXhl
```

```
4hvSh8gSvv5pbsh2AhzZhSOKgllR4nfdDeicBoNF39kAczSPwIOOEbc8FGdDbd2AFNFDvG9o
JGKNE+o8ikzLmUogCvSjTp0CdcVFAsmQrQ6O5CyS8gL5tHSgHIktwUxa5ppWj8b49XmITk9G
1zfw59QthdKaicObjCvcBmYORNrGppzNsRk7iXtXasvyGONEckroSuWTGUPg16KFjtZtrvSb
lHBRqgo4vl6qAooSQF1OpNb/5wBzBsVw4sdTPwG2xZ6Srw1SVInlDVCFdo9k/q+nAzm50dRg
UOoHDl/J4mrZp5y4nFwTeIwna0YesfZjybgQcEoLILkR4VQy9MgHYF2EJAbRxnyRsUIBhRZm
hV6WskF5GmX6ZkX2iOWo4p3alNT5Y5MlQaoK9dm6OJa4luyA0UEA4WN/G21A0WfqzEhgJ7an
IlU104QnoDya0+mqJfxAerYYmFT6FvZ8uhruh6FQ1QUCzYCtdH6RqoOumv5dlPXIab2WTATH
sPkES746xaYRcJt3qUSz1GUkDmd9P8Z5wK2eNm7z68R3bw0LyrbhXYFhDFYI37arKQ+YfYXu
MmQkKyy14OKINAS5TOF6SMZ55ahm9sjjoywEwFXpbCJO4pO7YdUUOvPn2guuHQnLd8GjJmQ8
W0qICgHIVyhr8URAAKbcWbg+cdBbvkWQ9yszz86uTW3bFpqsBMQjOOhJMGTeYzHrxTCNtn0z
oAQBlpDeDX1wq0hgI4RS0Pbz3o0L7QkVqUg1ag34YF6mY7HC+kfm5Ogs4YLrVmx/EAJjS70M
eGORJ22IVnA/7UeFnv7j3q5ZbIhMgB2avrXQEZJAJ9NyVRNAcd60hjlNYArj7iaS4YMGACAQ
AYHKoZIzj0CAQYFK4EEACMESTBHAgEBBEIAfbPfqWQAs6tcNrUTlvGz4AvC0yS1MTufh+TBY
QvZEWy25jzS0xieI4U/rDU8DwgkYnKsVa7FPZxN9P4X608E=
-----END PRIVATE KEY-----
```

B.2.8. id-Dilithium5-Falcon1024-RSA

This example uses the following OID as definid in Open Quantum Safe:

https://github.com/open-quantum-safe/oqs-provider/blob/main/ALGORITHMS.md

```
id-dilithium5_aes 1.3.6.1.4.1.2.267.11.8.7
id-falcon1024 1.3.9999.3.4
```

A Dilithium5-Falcon1024-RSA public key:

```
-----BEGIN PUBLIC KEY-----
MIITCDAMBgpghkgBhvprUAUGA4IS9gAwghLxMIIKNDANBgsrBgEEAQKCCwsIBwOCCiEAe917
A5n3xWbiwXKbeSyMbvRfnoPBlVHZ7DWje8aT9Z9YSWG4+l0K5gAcPqQ+wWBuIAL+S1HgOBPu
yS54ci6fATlQ85kgWlqgqWppckTX+DVEcp6PW6R5hGAjuCjtPDC2JfL/adZ94s6m7+SuQ22M
QpkYzlwPmpnsyYsSHxO6npaGHpvx0g/zigOVs/wLHmihRK8IAzngMkzue398NvpvEmdv6S8l
j9M8jImxDFlfj4jHCICtQlJIiXYjr/Y91LViH+XI0hH2X4BVmh+ebamMgjBMCO0JdoL5U6MR
Rz1N1v2RiEupPpYAuqtyv/p7CRWQ760U77buCu0UppUPTNHis7FBvssei44sxv9atlFebbIp
4+LqEycSm/MLBdXQ1ZJx9pMno4ttCgmg7mR0Z9E5hcPkCBYyejXnm8CtoEmSkt8H4+PssqtY
mBCenslKWqq62IRzPtCrb4b+snt3rPDhrEre6f2a4lpy4okT35H9ZE5pZd3DJPRMwOfRBG06
sCh0pS2+Bca/68Q1FqLikogxa8DIGMtzKRJzaTACH5wW6YqHLwdNSYj++phQosygnsBNGaYP
OizF4jVjkET1Khjjj3/YrEc4vQ3MEv+4eBeN4HPLULE5S+e7i1cKOiiSTLrov1Mb0XNrAZEj
O/t66DwLZARqr7BVAP/qgUu1sstbz9GkZwssJ/h5ILGALjMSkX/SAS72l7IBsWDq/ozcaJP9
YJXyLYbOQKDt8xcbkDOg+idlVChzYWABvfAT2Sks50ruL0Ttf1+bbGKezGYoUyBM6rPHrYgC
NbVJPfTIOZ4LFdG2qYZTMSvcMqUYGzN7p335JGVtplKiiXBjHQbhlNFna+pd5VCJCTvYh+Dw
w02rU8PzFrH+Yr0vcXa4tzJLfl9WNYAFf0pJxK63OGGxC+91PVQMorz4zE4WAveIRqXYc8+w
p/+chFXAfT43X85TfaKEbI2FjfBXkSXKRDM4RdzHwlSsSC/gNA7Uy7phndqxrLI+18wOVXtw
11Rk/RIBXw1HddPk7wXYi8UQ2DGMdg3zNdDk9C4H0OvLttX+X9jynJAcoB5ioTb/Y7KNMftU
tL0IuoJ2A/+9pZzRtkdNBmgw98QgtadfLnpnvQX8a00HpoOi+XIk+buA5+zXNVKWxKznF82x
qlccDjkgQdYFYmr2Mc1t479WBRiSrlC/ELpnm7lCA7wLIUO+u6SFJZxcfS7IiDvehprxUd0B
8fvNLY5qFZq2LtzBk6RegRrtQdT8LyO56wF9DSfPx0wZwk6eTRII3mTbbpzLGsZXKeSBPVRv
5CXQeZDKTkYt3Tn4vcTbYgjf7xpNmNMntB1ozG6g3SN6sCFkAr9tEbkeIbI3CToMGdO0PpCz
tmhY/GVmpsC+M0k9Lm4yHg2Ju+zwPskaHGruJZaH+u3z03TbSE3Zy4Ksku80mBb/x5auJgUf
GkS4pfSCSkHYf/AJzDOMIqSJaTKrmSQ4yENQNX+dAjfL95LHzYarjIOosmLOW5+gtxh/hyKx
oeg1ZNKbP4UHTC7RpNqZaBQ9w80q9LiBeD1lLn5ZATWDYleWtRvWX04hiFsduNwuSad7lTSd
8CYfAM7/3LouTMrlzv5xva+5Ununeb4QYnrSOeahi4eIolMJr9bp9vwrxoMVeD6vGskL3EEZ
R+te7s8xtHZFqMhzI8PuvADaBZVHKU6gNoKSL1zcY/fDK8jH/lLXRjbNAW8zuxrRQpaGaYQe
fVFOPS5uWdX6o0F3RQbTta3o7lMiAytwVpyMB+ubHk+h8HhgaOpVCIXR0NIC6tbIMkFj/Hln
A7AqBcmjq+pGKYpxFKiAMR6H/Am2lzDLmHZHhHXPU+mUn1+4vcm0ick/X2jlItm0d9g9zoDR
VBE4TqoA7i/tiKrNmsPzepGLG5XRnia7Ob/Y2y8hKe6e0pv/4heatS6W27oLbq6HladNRwzd
M6cB6uqzliQwW2BFctNhQXqzmbsmqxRNaIqbMO2YAxVTTHGD7UFMtPlbpCUkoHjn5wVaQrOO
wokF1qE5Lm4x4fwqDUEnn1IKNHvPMTrCGv2Vm2zFGpMIpGLOMRYTc/ybNORLjT+jNaS2pODB
uJzmxzoHZ9VcGwtWhVi1iGreZdADwLjOBZn/wZ3r027FK2iNFsOmf15nKno4qH63TWLb5aPA
0xeOYljjxcEIn0Uz++sRS+aRSvE7CMCtM76kWVs9muGguf+Dz/wE6ig5WAD3WdO+eJ17hkoi
ZfL/Alm3+65G8DbbOa73pZXErvFBXYHqj3x96mwwCETADiDiHt8wxD4brEVK8qSgZoYtYLNR
1ZIe8J88x7NsUKGHQ8SJ6aWGnc8VsuBipV38rDsG+f98e/3l0j1++vguiURp1B+sOco1H3oa
Ajfq1On1rXwnqPJ/yj3OQPlEhX8KmJOpcTuSgfhFq4uhC4I9w7VvsVOpv+FYr7tESGpPmyYp
2r5ZYlwEMBsdeD6myOK45wSLX3g+xDP1GIEQApRD0lwlvlirqPAUgdsWvI44ez9ZSHuBBW3d
3iFH0HJqXyUpAV3MUY3d7kTIUhq1Ji51nKzFEAG2dAq3oUDBVxfuWaGwuCl4RE6Pqyynw3jO
XePx9ohYUTJLfpBeeD1smsITdyQku0bC6Iwq28NZzN8RqHZNqPtat1gMGDoCJhj4m/6cSvjs
BrzJARf3pIEKCsMgobbkQ/j1woO5USYe6a4G58SoPbQX4bFMmbIxPpWhxw475HQbJMrzGnPB
RRhIhIYona//Y9/8qY674cDSQ+UFkJ6ZaW3oB8M0igW3QJ5mcsT/yxrjXAVAUh99AUfANjXr
0xO19dmRk7I8YdphiW6JgkebmG/SyACCFSnJ3jHqy57hk+RWE6na078qRkdOPLdIPXyJEuy5
qg+T/PmeBrH9BJWqExALXtoyNUV+QvUVBwfFU9edLqwAumk3jx2huCzw/khTwsckM+MQ7RS4
Yr/DPZM5vS7GErRKIomHZrqH2BED4SKT9k6Z9vjBBPAH9HnOeEIt1yXrS+kB6TKXVOf5vwQ3
dg3hurYkg2zkgQ76hTReuY3jWQl5XKDEJYkhLctCniZYdr7GB/Oiaw7f/bCS5uvBmdRYVhgb
zPZmrSCEEiHX2Do971T66JwHOhnAzVBbIq8PM/2ffxMLYOtdFj03N/RGdoBW21JXCl9igBdi
E3ACQAxxFfOWsRVvO6TiSliDZYaS6ms14EOdR9MsWP5VJjd39FzXx0rYjRPHNcuZnc+a2Glg
OXb0jBe2Ype86z5RqrLAedpYCr0ek3XjwOVZk7gvKMbrAf9S4EJKSIHZUSKEnZHNlEQUlNv+
IIHDzAHBgUrzg8DBAOCBwIACkYmV6hg0H5ZBCPAiUwsVRS5UCOpTnRdVVf8wvZlAM7Ttesjm
```

```
qsYbhHyuF3udFASyaaVT+p3dM0qzx7NWCBSgC6HxQmWQbSzwtxfzu/mS1ZEUWDG9t5H0QcRX
rLKSGdCriouqAbAcUENRYR3kT1WV6Kst6l7bkCK5Yi7R0+dSGqSwTnZrAll2CO6YBcU7teze
uaDAqv3ukSiqAvg3TQC4enH24S/1k/YEBYqoOAdRsDolEzB1nNoyPGTvC9YisOGE2mXoMtat
star4nAq2ryCi49Bic1byNNkcFyZs5j9DzazcHo1yCtHdG6pF9EKWp7K4wbRIF1asF71w4Os
XWnDglZwm5dp1EcM8mdvUrEKLlOTRC4kXc5DYdGRynDFdcfAGnOwDc432+MUbkorJVWuu2ms
i7e09Np2hhhV+h/xh9eOApL0o+4YrrWsa0eFRL676T92wxTWcAXV6prUN3b6r4DD3A8Tviui
oNQYICziEJILYabqzpF8F0ePVyFZfr5arU+bjv1+yfOpqTcxMnEeArqwjdfheFggvNF31lNh
G4GdRjMqk6hdHr1DTteljubNVcDem7HcLAeshsaJkUMdGaHEbqdjAbCbeMRh+r9UrpuqGRV5
n5KphH1aBx4tmGlglTMMmGvoSjAwpB050TYbBMLHlLCWWklfjIZZbqaelxkuzsc6gdhPA8xJ
k6FvMNwGMa1ix4jNj4ZtUGzhwlLiRoyUNKpu1pmgFpaVw9hoirTBasTNDZnraOjBUqQRpYcW
u/k32E6IiEYC2wFKNFWvicogMrLFdhIIk02eeZnJIu6mWaZ3p2LK6p/FnFhtJgq+LbggXIIB
u56+Zqhkwm2aDGjKv9AoFCS1FTgim5bJTQ7t9QKFAUodFvdiSHKH6UnXjRExbp6TRbruTpeY
RR5EI5PnQuxt+MRm5E0JgoEi0QxTOWsWoNOtn2tBVv1yipESLsQWCrlel2X4sPSNlh+mfFex
KGMmJoXAsGXrJl6TgDxwo7PqjBJIyfBJgOaRmr23GtAoSiCGguOz4d5RHNLqJYzeIDr1qLEi
Kt9eqDOKJRYrn0OdLLBxWc17npXzYZTRJZU+uOjWL8qqQCzDWdmLGWEoIiIpF1TavaAZ2XGk
ViudaYgqiRxpXazbpxojNofBV8YoycOXRgelhMKgnFqS9vlaFzacyRHSAUB62CpiAGWUZIir
JuFR80TlOINR27UPLR4tFkIle116cooG4uooNBrdCwbZ6MAIsVnZJiz8maRE+jYK5EBMZxEI
LEkxRgycvV52q+/qijZvaLNUXr5xqHrADASX8og3sdHDdvWk2B1dnqQuaRbvjXKs/gIUywGb
LeJXBp22qOFiOyWmsuiOyDNuRNlXc2YVvDItOtorLbE5ViruDK5YWdDKwWOUsSKrqFzUCrBA
2ggWGRLeC6q3FvyVaGWYBRRvqYtnMhZiRnaWqACbbSRZzp6IlzlxxirkH3yKHdFpm/eT2pcQ
tIjZ41bzjnxptQEYkGdt+xNL2a1EXK+ZqkYsmeRqYY1bdwZIurl6qukKRqSoTZaVJ5m7t7/a
ceAvxIsBm4jpRiRVKbPjoYZDbpIeKgGPCohREYdflk8argu7ECVvgBX4M/8DtAGVqzjb05FD
baFuchuoZ6JBsPxOK2RPtmYIwZtfYTMFIZDoW4nzay0DystB5W/MU9XdFhE9TrRlniJPXD1A
pUFqx2Mc0ZM+ahhDiAaolhGigYtzoIWNI+qJuo5Rz7dpA3skXTa7GccKiSVYotpoK0qMNkJO
gEDp06+EfcpwICnBALn0ifoYDrT+XhGcrfSUeedrXej7GsmolxJb2r9y+2YThv+0MpIns6+T
IB/7axbg0cmsKLt9guAQ4FejKZrTxilPGMq8Srnv1R8e7nP9zrc3uXN76UcTda4Op/1gfFnN
yhsoxaA5XkEmUI6Ho5rcArwNo3XcqixU+We3ToBv4gpksU10YwRgu9McX4+YUHnLK3ytgndE
YYmEM6NOtECd6RVWEkfAs9fGFiEcwPvDY1ebRtKQEuNnvGcFY0e3i4qVlbPXGUrftgH2QgUI
RuDVCyNheYV9kX9EiKdildUETwVvkULCh1DpwBkpm994RjL29FSOHMzLYAbaiDOKiCQWA5l/
YSk0CVwiAthxipC1GDQa/ZYRMwB0aa0p3R7CnwcAqdHMgZ0a/CNoW7VglSJMKaN7iaOprUFM
rcG4io3mzjrZ1Jjr2387EO3p54fFVSXN17HDxD1Qh5atqoNNCaX0+hz/uaKmA5LHQosc82Fh
JcB5Do0eiHfQrAwggGiMA0GCSqGSIb3DQEBAQUAA4IBjwAwggGKAoIBgQCc8x0qq2jSuQhAV
LJpeYHeUa/x4tZlLUhbTziOADVcmHwJaupEXAzViCqoQ0Nk44ZmR9sdDPatz/Pil55Ssrv6N
GHV5HWkdU0AxAtcUD5YXQbNCNEDVyAtVKlbYBokr+jS7Ed7In3+5fts5tenP0oCaAD+gi8zb
fFXowZMzCZByQaz2JkvQZjS79tx6YxEHWeTdnmIUpg+ueluRGRw2+hSAFBdPIZXLKDbSR9Wr
/kEBmrcKZEbEIyPnyq4iHvivtpi0rgqGjprGvFo2GTvoJ76GOdZEutkU0et+CEbtmexUmySf
o8EufEf293W37DggP6aA6ipIf9knBIekPjueWaxOY6AU3aG++htyVL0rlgcxwLKFEqXIM6Wv
G1mkcVNsQSkFr2Z5+Bn/+vYVBJYx1EDTqNj0/fZY1MasYDuWXImx83iCbmBIGzZ7BCAgFL9r
iKAw5IRVQWXCEX3g8x4UbQb9L0gq1MnqxVsn5g5MjpG8MCAwEAAQ==
-----END PUBLIC KEY-----
```

which decodes as:

```
algorithm: AlgorithmIdentifier{id-Dilithium5-Falcon1024-ECDSA-P521}

subjectPublicKey: CompositePublicKey {
  SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: id-dilithium5_aes
      }
    subjectPublicKey: <dilithium key octet string>
    },
    SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: id-falcon1024
      }
    subjectPublicKey: <falcon octet string>
    },
    SubjectPublicKeyInfo {
    algorithm: AlgorithmIdentifier {
      algorithm: rsaEncryption
      parameters: NULL
      }
    subjectPublicKey: <rsa octet string>
    }
  }
```

The corresponding explicit private key is as follows. Note that the PQ key comes from OpenQuantumSafe-openssl and is in the {privatekey || publickey} concatenated format. This may cause interoperability issues with some clients, and also makes the private keys appear larger than they would be if generated by a non-openssl client.

```
-----BEGIN PRIVATE KEY-----
MII0cwIBADAMBgpghkgBhvprUAUGBII0XjCCNFowgh06AgEAMA0GCysGAQQBAoILCwgHBIId
CHSB73Xv4MMDmffFZuLBcpt5LIxu9F+eg8GVUdnsNaN7xpPsrCOsO/QRu+GObWvZMyX4xyy1
3gZf/HpZhmrF4PHSrex35iMpaLkpu4IP+2fjaA+NXYOO0Swi3DRp9d2XxKwmGDMIkiKQQTI5
I0Qh4GZGHEJxzAUt4wCsolCtmDKpEhkFJIEApELuECCQCbRNoWKBiHguBCSgiijNG1iNGKis
poTCuEiUBoAJBCCRCFEbQCChpCWJEoIZR4pbRiQJqI0IxQmBQJGhyBHiCAHhsoGYiEGKtoDT
jtggZmG0aFJEkpxEhASyUFCzCQoQJAIwME0CMiAHQJGUDpyWMKEzCqEkURiIQMgkBARAkAyT
6IiEVESE4CR0FkNFCAAHGTFkYMlkibRjFZmBCJkmGMwAgURZGKSDIjMIRaRohIkGyIICpkNg
kXTgI3MEi5REmlauIHQCGTaBDLglIgYmCEMAy7aooQKNShLwgUABCQjAYiToEgBJFDcskkAs
BnEkGEygIIVARE1RxilTwo3LNIURCUxkFFBJpFBMAnHkhIEZIJKhECqQGAoSAG5SwAkhiAgD
ZkmQKwo0DIzBIsnDCpiVEIkqgiCDjhGjYMDEJIkDQEm7QxDFiBCDUJGxIRAFcxkUisGziAIQ
oIMxGApkiaxE0IwImAKIAcho2MoGGLhEkTRCEcwEVIGHJICC6CtmVhNAbJiBEUuE0cmI0jGE
WQUFFLbMJICBCmYhAkCRlDLOGYaEQoMBJIQJmULEYFEFijTqAgKR2YLBpISQkbTCARJlmyCx
QgnkKGwEQyLcKImMiHHaRi1jRiokERFQlAHkOIYkF0EhxojJkGnZIi0QtmVUFCgjQykYtUmL
LRRAKh4gLRY4EBYhjCCmgxIxatAETiTBaQjKjIAJUOARjBgWgMIFTSDAKxWmQJgkBIwCYQGa
CTgGCaMmQMhCRaxAQJRGHECCUiNoHgICEkFGabJoGAGISElonARAEkqWwcNJLMlgwRomkBhy
i3Coi0YBkUCxWTYECyAJEAEIE6SBFAjRmQCpYXUGJFcIgpSuCUZFEAbM2RcAG4RAlCUSGwSl
uG3TsggZgAVUSCbbIDIExZBAIgJYGCrLuCUaQowaNYYYNjJZsmmjtCwjsoULIArRREpKmCwQ
IElETAEngEkLhQikCkQ2aBIEgw0QZQjFKECTMEAYjhQXbRi7BJCwaACrcRk6ioHEbsgEjQIo
KQGFJUwCFiJJGbtEXiliFhFChikiEZJ4QaIAiiGFHBEnEaJYUJiDCbEg3ROAkbFWiDxGHSMj
IrABpGkRk4hkyUZGE0KE0jBMGFYAoWQpmzBAISCIo1JMowUoAUhtEwIsIAKNFJSBoVEAkIcB
pInDBC6IGIKIxGiZRiCItAkCSEZAqIALRhHbEEwQgwgjSCDMFCQEiVGQgEQZuFAQCBERFiVJ
DKEhLhG3KAkqUiCgiSTAkiYgjhDEhFEgLoG0LMo7bKFLUKCCMImHBNkBZIEEYooxCJi0YqWE
wDmWFaAkLYuC3RphEaQDLiwnEJljFKJIoMEWwChm2UEgUkMBIZsEACBXBLNAUSlwFaFGrToJ
gHJhGSjogQBGXAZEQ1QqCgZsiGShAzAhIDiREEZpXEREDKKRIQZNSLIpg0MowyShoEBJyXAR
SFLBxmwahEkEkAzjQIoLppCiqIUhOAVTlDHTgklaCAnaEEnIKIUBQQIIQ0lDMJJLMDCKMkAj
MME1homiYFEkCgTDhMhLLspARRgLZOJLboElgBCkKow2bkgQCA2LSlICTQiCZFpHbGFJiBCa
yiGFAMEG4aOQIQg3FbolEYA3IAmBDbGCHYBCKaAGiEEgoRyDHJhkEJQlBRIGUBySGCso0JE2
gnQEEaKIhEbQy5MBIoiRmGksoADJUkLEIkbQkAIIgTTooUAoGlTuIDkti0ElREJsi3cqGxjK
uAARR5KhgmQEBgVZBA7RRIELGVBxX17fATiG6BLmyaJqAkS3lqIoEswgKUAGcxqaDLXP0x1e
jBfLWki6E15q5BPHjFq1dcUHuzlQrLs1VwdYg+ICy7UcVpyphBmdN1tdKm3OaxdMKrUAVo/L
xGeL8G7fDFr4K0JS8Ytul23OLxtGiZ4Eph7Hvq8xdaCwW4TFPe3LgdBenvNQDDxQj/LfTESo
R3+zz2/eGWLH+2Qye8mYroAns2UEsrvslP6flBcmPO/VhkIWihsSw6uaaIBW3a3mODFGfRJd
hWRv4zQd9YGGQmsPgNfW0+HS5V8mB2wcJy7sL07w8PRfLbY3CMgo7yNMA5a9xhMYvFckzhpX
Kw3DLVw1sRe7WJD3x0/t7GLlgDMqn6Jwo3xuA0Az3jVpx085CarqwUObgtTSF0CRTf3gJcKb
lJ/5i60xWoZzseX3K+ReFWUgSt5Xk58PUukjk7QspUUMBbwIuPnGr8g0bRilgzWqZqHiIasA
enH95FOGyg8SazcmXiIS5kv1WpMmb3PPrCMI7oMXwx02hNCodaY1vhalWZXE30fB6aOLMdPh
MFX8G7mUKcU1TXC85JaZh71OF19+iCMW5v1KQEGKH1x2Kv5wqOYAF8/R32dHh7trnQLUrT97
oej4qUxiQbn2SqOsWNjTWQ/VK/94nMdf0P7EuCuTACg6OlANb3TxvfJGJTE90szMrpPDSuGs
0ZyYioZcFMF0u1bGIXreLSAQYueZB+UjlUE9CK1bwVkZKHMkw6/rCuHfy33iwX03TteM3Kcb
r0jMubPPqZKGu18GqcZGjcoiXMuT8W4jYznxTyaG/3EpNsBnGDTgCaL5gKjjitNcbCQviHml
68lo9/PGiEVlYgutfEcz+uyxQYvBdH4TUIkZZZfRtstmSO+Xv8w4lqFvhUVPc616K5f0+N9S
N5dyqXUN3h1xLTVQ6zYjhAglr7prhDX4/6ympjowZQDiImVrIT2uxOqmzoLLUlGMpEwWdpUM
AlBFfSASdkNSGzProELg6yzxYoLGruFeZPkct1RMrhMFgJYy/M6d3OEnDv/+937FTgoC0X8c
o2FTznUqDPtk9b+OAgMqAKcXBWzHPFKa/zAMJWLAvdAccYWdGbpH4GdpR1KkFQCPmHVPkf1C
/UgumocQ9JUtEVefOPxy4Ax6G0nRPXTYY6SPTGeNkTyOpZHbHV20kHiOQCwXVtv687TJWK/1
I3B4niZ0nRSRNZAzlK2QRtLCh3PV9cNo/zlUw0/g0nNO0WR/UGO+wcQXmlvJqHqGO7CsAJbj
grFoO1bo50jUx1BDgXHiV2qIic2E3z0hj0yRpUKCuO6UCrbXYBJddse7pCMuVL98ir2qk2kE
9B8AiIpQToqDJ+KWunrzhZFY9zJvvOSm2SvzrmoiiAyKLIp21QGilufwZM/BmsJhH6a1BiuI
```

```
W/pMMH+UYaLo7FARLS/PdJjv4R16ZXQ0Msyb8UvHssR3bAD7kWAlBRapR7iXsame2gr6oCkj
e+puc7i1gno3t9yt5Bq3znd0SDwrxlyJO6GmHO7U6CO10InxmJRbr31CsjotGl7l2aQB7ZWo
NxysBhAsHT212lSUK8XpwEgOEOQRjuruo5KUVUsbr8i1tm3gRca9fTI4CevsBguXY7C6njOA
celdwYCstgrKMfCnaz39oZpp/MfYgqQz2U5Q9BKPKWTscoC0UoBG4pS0W39+eT++yKdLswP1
wORNX+cnSCp2zKt4n7c9yypdZri1SuQekHnMioF9y+ZdlW27Aap4xM+SnzEe7A6W8hOiyoFJ
GYdGCNqOVzMOaiS6W4ZLoJrFSSxyoqFI266N5IIPosRftKGPLwtKqq5WMttCjGN/zC3X2GX/
gp0M32X9rk3c4fXeVbjEBteagQLelqArRM0sANTWt1eTeD/fYCHGQzqV5wopIrgnlduGrlE/
VKPZuSW7kcyq2CpGeWX0cIX8ZERWXYLjIS32HuxyiD2zQHlldnOGaW1HFP2zg330/Fl6GWYF
jmWeeTuYG1hjsSnMUS9rf49ckwErF/6mO6aqYIVVkKVFyT7kS7F4mHjVefHuAycCvh8VA21r
fCTaukI+QFuAeshxLk/lE5+bACZ2X/vq1Q0UBrnBu33QPn5ptT9LEBqAgAx31CSFsS/oYs0Y
dR9mofc7UEDo2wjkiTWSH0/a/iZrT2taQuzJth4CciEqtSRdV7Y9agTY1kx0+zRRvvARTvcQ
jH27OzvBNeHxhOrZFzJHEn3w5jBRVMkARH0Xs3inOueB9e5DY7MC2FEgBMnAKzkD31lKMlVH
6DB1cDfRZg5Lf/fBAB55KQK5K7saPgFuLnIo/tIxVAq47/vC86HUd73i0aWulhHniz7ZnkQ/
Qc+s3/r3s2MHRzz28LWphcTl0oKvfP35hws4+vyYEUzNjClPEKAfFZMxf/hvy7KMHsGn+EzS
svos5g/vc14hQQ4TKOQaV+NkisqHAYUegz9sWoUK72mMW9XtjvqApjd0sz4KQnJbDJhNUyGK
ddda/Yi9I+MNsm819K/hw4cfV3XgwxLErncWRlON7MCYT4G9RyaqrizhOZHysoAfvLHwlC62
b2fg8y8C1Ab9Tx0B4FDGyo/0rNtsI6X/tXpNb8ddCPy0OSrMHoW36H9sPk5o2aLezKd5iDVf
bB1aKDmfixrnibDhOrMwQQFoobFd0NuhJWpDE8/VbgozoQhxt4RJCAJ3bWiJ8MmbVNeCFDUW
rAnk1HtZW2CpBJZWiPerC2BK22FRpNSgaHieWGEeIao1+LsjatkSc63J3lZ56WWehYlQRJD0
yhDbCv5f8j+a8DLB7ABUWoTlmqiKXPbAUIPgXtNawkK1108zjnsaUl51CJflcS9BO36CllFY
jwRmComIqB6r39aW5meZYUUBdVOLpf4+WYh6aj1VjCqAjq1VQP2ongdeX+JwR4lgsrhxsXk4
jgbz24djZBwkYG2xe0MHkG4NFbgmUcNZ725EoLIHx8Up9YJWG5J/PlAMn3PFz3xuc1S3fDbs
nHhmpGOPg89k92NmC7BoXM9XbN8aWCJhfiq1J4tu5Dlb6Q47uTmtnAsr3e6a2dhsEIaIIqR/
yAxstEHWnmq9Hun23ldFAN679kJLpxt0tMLIDDT1GOH4twvg8qOuHDMIhecS3S1N3EwYaMfV
PycwpmH2Y3RBZ0Cc1NFAIakxv6WgBRlNOW8pAiBFLWfYQRMUNlgdDzjQTR72vlwgC6J4Feta
9YwidcUTXSaB3he0x4WpVXVO1ZA+FLLoqRzDfU+rgezTN4V+FjE+ADP1CiwLTKWZ9OoaUEoV
b/MaRKAszT5tY1rFB1HUdvRf5Ve66XeEh8H96+RyU364y27dBBWG2w0mzNQhNUag1R5JLHn4
hda3bzzHfciFk5j50TA0qFIVYTEY8yosVaEXSvsetujQVPPFSwmbx/pYWH0MgYOPz21GgoRA
94EW8oiReQ7QfsG8F+eqpmlIKBey3+w0TJB69U9IWhJ6CyhfIwCoLA8pwxHjiRLC6evWnBjq
hjMeAvuhzbB7KkrM7vDUBICGWVCycsYb7NJYhw4qQzCKqsSCT+Fb/oSALT6Tray+eyNloWt/
89wc5DGlzkBvu1BzzquGoC0sVqlEi+kZ0t2rqfNG8KwyQnm9PD73wB6Yg5VfAxamtqgQu4/O
TaWrV2a7pnctiVCm8o+3vo8Mys9oUgOKLF7I2Y3uQbAMyzUKLQ5XEk6DE0roaKk1qXnl/TRR
PEAI0+pGQz3eYOZyyAnzzPmXwFaywEYcla7bxAVtxyXN8hDxyjpzB8A4UcMiii4HLSXpeHCe
1vjM23G26KNz1eqTyXwuMoVtP4JMIBpkwjMpm8LkN2OTSjcsZxVoak8FNZClTzw7gS5sNXLd
T3SmEUxjx9CSqXZoq3C0uZMQf4KinSDejZFlzOVVP8/Vo03AiBD2ykRx6K+pXHtYdQddfX+F
59A/IZvcy74FaMFR6VMVeKMZwq5CoUsbd01RS6wSrljuLNcO1+sprIIqwbqub0DRplSdNRGG
23KUvpS4nd3lURvdPITEQflMM1kgJE0CXdT0zbvo9DWsbvGgwdBN/Ubs4D7Ug4p1GdupuFfN
kN0g6xaq4lqQOgYAWtaYzrclvM1kM4FWvJhahfhSVPsqLOdhjIKDT05X2cm8pfw5lZHzuPUV
maOhHpbyc7ekqO3mAsJQN81X5VYCRDiO81Bb+DVXHgC+aDmoiVwM3ruzy/VqLHio2Kzg8q0s
cRtNDDCXS1hyo22w86ZvTIOKjMxLqYKFVcW4jRJe917+DDA5n3xWbiwXKbeSyMbvRfnoPBlV
DWje8aT9Z9YSWG4+l0K5gAcPqQ+wWBuIAL+S1HgOBPuOGgyS54ci6fATlQ85kgWlqgqWppck
DVEcp6PW6R5hGAjuCjtPDC2JfL/adZ94s6m7+SuQ22MjF6QpkYzlwPmpnsyYsSHxO6npaGHp
g/zigOVs/wLHmihRK8IAzngMkzue398NvpvEmdv6S8lGEUj9M8jImxDFlfj4jHCICtQlJIiX
/Y91LViH+XI0hH2X4BVmh+ebamMgjBMCO0JdoL5U6MRtEuRz1N1v2RiEupPpYAuqtyv/p7CR
60U77buCu0UppUPTNHis7FBvssei44sxv9atlFebbIpbXg4+LqEycSm/MLBdXQ1ZJx9pMno4
gmg7mR0Z9E5hcPkCBYyejXnm8CtoEmSkt8H4+PssqtYsprmBCenslKWqq62IRzPtCrb4b+sn
PDhrEre6f2a4lpy4okT35H9ZE5pZd3DJPRMwOfRBG06XWGsCh0pS2+Bca/68Q1FqLikogxa8
MtzKRJzaTACH5wW6YqHLwdNSYj++phQosygnsBNGaYP6PkOizF4jVjkET1Khjjj3/YrEc4vQ
v+4eBeN4HPLULE5S+e7i1cKOiiSTLrov1Mb0XNrAZEjYfEO/t66DwLZARqr7BVAP/qgUu1ss
```

```
9GkZwssJ/h5ILGALjMSkX/SAS72l7IBsWDq/ozcaJP9ZtpYJXyLYbOQKDt8xcbkDOg+idlVC
WABvfAT2Sks50ruL0Ttf1+bbGKezGYoUyBM6rPHrYgCTTKNbVJPfTIOZ4LFdG2qYZTMSvcMq
zN7p335JGVtplKiiXBjHQbhlNFna+pd5VCJCTvYh+DwHnHw02rU8PzFrH+Yr0vcXa4tzJLfl
YAFf0pJxK63OGGxC+91PVQMorz4zE4WAveIRqXYc8+waNIp/+chFXAfT43X85TfaKEbI2Fjf
SXKRDM4RdzHwlSsSC/gNA7Uy7phndqxrLI+18wOVXtwXT411Rk/RIBXw1HddPk7wXYi8UQ2D
g3zNdDk9C4H0OvLttX+X9jynJAcoB5ioTb/Y7KNMftUVLstL0IuoJ2A/+9pZzRtkdNBmgw98
adfLnpnvQX8a00HpoOi+XIk+buA5+zXNVKWxKznF82x4k5qlccDjkgQdYFYmr2Mc1t479WBR
lC/ELpnm7lCA7wLIUO+u6SFJZxcfS7IiDvehprxUd0BQEN8fvNLY5qFZq2LtzBk6RegRrtQd
yO56wF9DSfPx0wZwk6eTRII3mTbbpzLGsZXKeSBPVRveRA5CXQeZDKTkYt3Tn4vcTbYgjf7x
NMntB1ozG6g3SN6sCFkAr9tEbkeIbI3CToMGdO0PpCzXRWtmhY/GVmpsC+M0k9Lm4yHg2Ju+
skaHGruJZaH+u3z03TbSE3Zy4Ksku80mBb/x5auJgUfS4MGkS4pfSCSkHYf/AJzDOMIqSJaT
SQ4yENQNX+dAjfL95LHzYarjIOosmLOW5+gtxh/hyKxw2loeg1ZNKbP4UHTC7RpNqZaBQ9w8
LiBeD1lLn5ZATWDYleWtRvWX04hiFsduNwuSad7lTSdwVk8CYfAM7/3LouTMrlzv5xva+5Un
b4QYnrSOeahi4eIolMJr9bp9vwrxoMVeD6vGskL3EEZOj1R+te7s8xtHZFqMhzI8PuvADaBZ
U6gNoKSL1zcY/fDK8jH/lLXRjbNAW8zuxrRQpaGaYQePfCfVFOPS5uWdX6o0F3RQbTta3o7l
ytwVpyMB+ubHk+h8HhgaOpVCIXR0NIC6tbIMkFj/HlnM7qA7AqBcmjq+pGKYpxFKiAMR6H/A
zDLmHZHhHXPU+mUn1+4vcm0ick/X2jlItm0d9g9zoDRScxVBE4TqoA7i/tiKrNmsPzepGLG5
ia7Ob/Y2y8hKe6e0pv/4heatS6W27oLbq6HladNRwzdUHrM6cB6uqzliQwW2BFctNhQXqzmb
xRNaIqbMO2YAxVTTHGD7UFMtPlbpCUkoHjn5wVaQrOORrnwokF1qE5Lm4x4fwqDUEnn1IKNH
TrCGv2Vm2zFGpMIpGLOMRYTc/ybNORLjT+jNaS2pODBtnQuJzmxzoHZ9VcGwtWhVi1iGreZd
LjOBZn/wZ3r027FK2iNFsOmf15nKno4qH63TWLb5aPA/MS0xeOYljjxcEIn0Uz++sRS+aRSv
MCtM76kWVs9muGguf+Dz/wE6ig5WAD3WdO+eJ17hkoibdCZfL/Alm3+65G8DbbOa73pZXErv
YHqj3x96mwwCETADiDiHt8wxD4brEVK8qSgZoYtYLNRuhX1ZIe8J88x7NsUKGHQ8SJ6aWGnc
uBipV38rDsG+f98e/3l0j1++vguiURp1B+sOco1H3oa5dsAjfq1On1rXwnqPJ/yj3OQPlEhX
JOpcTuSgfhFq4uhC4I9w7VvsVOpv+FYr7tESGpPmyYp/OU2r5ZYlwEMBsdeD6myOK45wSLX3
DP1GIEQApRD0lwlvlirqPAUgdsWvI44ez9ZSHuBBW3dmPx3iFH0HJqXyUpAV3MUY3d7kTIUh
i51nKzFEAG2dAq3oUDBVxfuWaGwuCl4RE6Pqyynw3jOCSOXePx9ohYUTJLfpBeeD1smsITdy
0bC6Iwq28NZzN8RqHZNqPtat1gMGDoCJhj4m/6cSvjsryBBrzJARf3pIEKCsMgobbkQ/j1wo
SYe6a4G58SoPbQX4bFMmbIxPpWhxw475HQbJMrzGnPBOxYRRhIhIYona//Y9/8qY674cDSQ+
J6ZaW3oB8M0igW3QJ5mcsT/yxrjXAVAUh99AUfANjXr9Op0xO19dmRk7I8YdphiW6JgkebmG
ACCFSnJ3jHqy57hk+RWE6na078qRkdOPLdIPXyJEuy51sbqg+T/PmeBrH9BJWqExALXtoyNU
vUVBwfFU9edLqwAumk3jx2huCzw/khTwsckM+MQ7RS4AaxYr/DPZM5vS7GErRKIomHZrqH2B
SKT9k6Z9vjBBPAH9HnOeEIt1yXrS+kB6TKXVOf5vwQ3ch4dg3hurYkg2zkgQ76hTReuY3jWQ
KDEJYkhLctCniZYdr7GB/Oiaw7f/bCS5uvBmdRYVhgbABfzPZmrSCEEiHX2Do971T66JwHOh
VBbIq8PM/2ffxMLYOtdFj03N/RGdoBW21JXCl9igBdi8V0E3ACQAxxFfOWsRVvO6TiSliDZY
ms14EOdR9MsWP5VJjd39FzXx0rYjRPHNcuZnc+a2GlgPKLOXb0jBe2Ype86z5RqrLAedpYCr
3XjwOVZk7gvKMbrAf9S4EJKSIHZUSKEnZHNlEQUlNv+RSMIIQFgIBADAHBgUrzg8DBASCEAY
ACWvBF4gxCAIWyD+H4Rm38IOhAL3ihN3//g/4Av/EQfweADhSc6Ln/DGEn9fMQPxi+Hv/jL8
MEAgiLsABB50XyANvZBE/8fwfGEmw+EPuvB+D/BiBz3BgsHowkN/4OF8D/fj+AINfAMAQCIH
///B89sIPc0IhOf+AYPE/8AiA9/3TA5/Q/f+EIPDF4vQfKD/v9/zxegAEQD9GQIx/J4AQh8D
AIAf973w9/78Ah56IYPC2f4wiEMI/lIEAOhBsHwCCH4x/AHmRBHz5A+D/nSm2QPwe77w+CAD
GQH/9IDvhf+IvwgAHwgi6IQ/dCIPC9IL4BCKIf+A4LovhD4IBdKMQBED4QPj+EHfhCDv+kJ3
FsHd9IEnuA6D++eJ7ZO+B8AtgF3xN/ALYfBADwA7N8PgDAAAyb97PgG98RwAF4QRiD74v9CE
GIHffP8IgkB7/eBAIIeBJ7/f83oYgZD0Yg/CP4ui+L4QEJzoxkGEIAgL3gA7F8X+7F3/y//z
2YJPh+DYAhB3pQCD8Ike4QPRiKLogfB/gyC+EXhfIP4ikBv/r90Igxn54AB/+Dw+GEH+hlAH
/8BCiCIASd8Afw/ALPhgL7//C73oB7AAGwg98vy8ME3wH5/5fjD7nt+JoIwD8InxEAQAvl2E
F/4OjKHwiBB8HxgzkYihHzwTeAD4RB6HYgjAH4QB+HghB8Hn//9sIiBFz++hHnoRb6L/i7+Q
6LhPl8EJAAJ/wNl9/4RE4DoQDADvx/CHvDfF/YicKMIvCHn3QBEMIPh+Mo/jKEAuCIDwhFH8
EEHOBADhAB+IAPd7/3i+GL4we/zwPk8DgBeIL3h+0EAeaB0nP+H4fv9J8YBA774/A/o3+9EI
```

```
AIAgD38u/iHrwvBD/3Q+L8oxf8T3ygEH3wACLwwlF8IBeALwufGAA/i4D3w/D7/xAD8P/jGP
F4P/fIQIg7J8QflB33ycCcHgAEHYv/9839j6LoAB+P4fAGIRBBF/+j/AEwACL4H+i9v+/kKL
8HXgDCDo/ACIIQBF4B/EAPnAe+MXQAB03wj50Iwi/7xPh/73/fCMoO8/3oxg9sHRgwAIPjJ/
IAYhg6IQ/++AHgcGAwOD77fMmFv4f++AveC54P/hIIRPCyEpDBGHwgiAIYL7yIW/gEIAeDCA
AIqRgCDw+HCL4ya8LuAC+EfgC/f4vAKARfg+T4Oi4Xg/C+cAxg2IoecEbwQfILwgC+MPvdCD
6QXwD+HoQax7YeaCAIegHoQR/B8mxe8Q4h+8DYQ597wgB+P3QGDrohC/4YOg8IgBiAQARAJz
AIPQjKUA+86EOxCAAIs4CMJwgIMovgH/gPl2IYvC4AIdcJ8QQDH/wRjD4oiC7/YDBJ/pAA94
+EAReCLwAe8IH9A/4BBh5/4id+XngaH4XgA+HpBE8EPg/8D/i98II++57wum6EHvA+PQAfD8
GABPcF8JOC7/fxEH4YQ9EQQhhEAXQf7oPiAH8JBe//gvD0gpPj4L3x+53/weCMBgACDoA9GM
z0H+6/74+jH74ReCUJB+8PeweCE/uj/kAuD5oHQAIbpOeAEvRl+LYhF8T3/9AAIQ998e/gEL
CMgBBEAIdc3wISg+AfPCCEfxCCIXR8CL3+i/7+yDF8XBAADof96AIPDAIXxBAy0M1w8DByj4
bFwD+At/F69sSF/oA6QQp+Abn+SwjHAUL48kH3+n+9fMC8UUJyPXt9BkJAeQF/vbq5BfwMPk
vfAuQcw/cL2OcO/xvuHwEVB0UbFdYMC/sbFw0E+QjzCxHVGCkvHDf3HQbuCALy/BP/7ibEye
tv7K/gKFf3k8czbE+bbAQAk8gr1GObF9SUR+fUS9fQH9f8A+/0S6gcg/Bfn3Qz53BIJGBQKD
/ekM8OQAG9sH0iLy0vcF9wv/3v32A/EO6OEU8AXjBQDlLPPxDQ8UIPAjFR7dI9Qw5w/zEyAF
rDhPy7/nuBg0DCP8e7/8F8gP1BfQhDQAWC/7/Grb34gwN8+nLAwcJ1vfxISINGg8N6NX+0QI
vy/SYODN3l3+gTCAb4CBEGBv3bJ8cC+ST85Os1/+TZAv7w+xQg4SsbB+zuGsodFP3/BBn0+O
BH7Pfv/HCjkIeQQ/yw+8QkKBe0X8v4L/foB7QghARr98hYs59cH5BT3ACX8NA7eBwAr9BcQA
GNPNG/jkDfQdDRoHJBAazOH/GfcPAczVxC/NzyUXBxDtBfjjHOsE6BcxCN7W0gP29w8NFeni
v4OICBxL9CPT4DAYGFf7qJALq/Lv87BL0+gr5HBz9FxH5SR/x/ggL0Rrf7CIDI+YOFPQU3fL
zt+BQD3uTF2y0CFRkdIz77Ch7w8ykI9AL0IRP98PwQ5CYXCvQIIAHw5+nQFO8NHg7PBxsm7B
xUjDRnu6A0RLNYCDuUc5Pgn/gkM9BTg6QztGgLh8ffrAcAj6R/t/QEf6A8fDeXiACD3COsmK
APr8Hvv++wXuBer8/zbZ8BELDUII/SbsLOYSAiYPzeYP7xfr8gf7EwP/PSIK2OUABOAN4f4G
s+AntIfQN7RcMFO746+AIAvTyHeQTFjDoIuz34eIo3vTi+RXw/ADf2+YW3eQb7O/uEM3z5/Y
zfD/Uf3yMb6wU9Cw4y+NcYCyn+A+sJAPcjKg36H/3WCAX2FewcByH2JL4A9jALH/oh/RP63Q
ffk7CEFDPIAxxzWCvk2Bhbw8QtG5tccFxo8IAT8D/8m/wL06/AkAewaAODdMgr5/+nrFQINz
6Qbetg7eJfPlFCwB5Mr1//ci6h8IDB3h2AP4CUvs4EEoGBgP5eEG6RwD9QAl5+3vC9/7BAID
zFOD/+f82DPYZCADrDhUf5Afy5fz17hIIAAomGRAE5fEG/iUcDLD1zAgI4u7o+UYeGgMZ/Aj
gJ//r93ADy/dAUKBMLIDDqCQ345fvyJP2uHCb40xXKCPMi9zXvGRHg7On1KtsH+NURE97d/S
gpGJleoYNB+WQQjwIlMLFUUuVAjqU50XVVX/ML2ZQDO07XrI5quGqrGG4R8rhd7nRQEsmmlU
3TNKs8ezVggUoAuh8UJlkG0s8LcX87v5ktWRFFgxvbeR9EHEV2SfKyykhnQq4qLqgGwHFBDU
5E9VleirLepe25AiuWIu0dPnUhqksE52awJZdgjumAXFO7Xs3gOp7mgwKr97pEoqgL4N00Au
9uEv9ZP2BAWKqDgHUbA6JRMwdZzaMjxk7wvWIrDhhNpl6DLWrQn9bLWq+JwKtq8gouPQYnNW
ZHBcmbOY/Q82s3B6NcgrR3RuqRfRClqeyuMG0SBdWrBe9cODrGWKV1pw4JWcJuXadRHDPJnb
Ci5Tk0QuJF3OQ2HRkcpwxXXHwBpzsA3ON9vjFG5KKyVVrrtprFfJ4u3tPTadoYYVfof8YfXj
9KPuGK61rGtHhUS+u+k/dsMU1nAF1eqa1Dd2+q+Aw9wPE74rosYc6DUGCAs4hCSC2Gm6s6Rf
j1chWX6+Wq1Pm479fsnzqak3MTJxHgK6sI3X4XhYILzRd9ZTYa+vhuBnUYzKpOoXR69Q07Xp
zVXA3pux3CwHrIbGiZFDHRmhxG6nYwGwm3jEYfq/VK6bqhkVeWGMp+SqYR9WgceLZhpYJUzD
6EowMKQdOdE2GwTCx5SwllpJX4yGWW6mnpcZLs7HOoHYTwPMSap2pOhbzDcBjGtYseIzY+Gb
4cJS4kaMlDSqbtaZoBaWlcPYaIq0wWrEzQ2Z62jowVKkEaWHFgMabv5N9hOiIhGAtsBSjRVr
IDKyxXYSCJNNnnmZySLuplmmd6diyuqfxZxYbSYKvi24IFyCAR5QbuevmaoZMJtmgxoyr/QK
tRU4IpuWyU0O7fUChQFKHRb3Ykhyh+lJ140RMW6ek0W67k6XmCYSkUeRCOT50LsbfjEZuRNC
ItEMUzlrFqDTrZ9rQVb9coqREi7EFgq5Xpdl+LD0jZYfpnxXsahXChjJiaFwLBl6yZek4A8c
6owSSMnwSYDmkZq9txrQKEoghoLjs+HeURzS6iWM3iA69aixIj9ryrfXqgziiUWK59DnSywc
e56V82GU0SWVPrjo1i/KqkAsw1nZixlhKCIiKRdU2r2gGdlxpJIJVYrnWmIKokcaV2s26caI
wVfGKMnDl0YHpYTCoJxakvb5Whc2nMkR0gFAetgqYgBllGSIq+iZibhUfNE5TiDUdu1Dy0eL
JXtdenKKBuLqKDQa3QsG2ejACLFZ2SYs/JmkRPo2CuRATGcRCCsACxJMUYMnL1edqvv6oo2b
VF6+cah6wAwEl/KIN7HRw3b1pNgdXZ6kLmkW741yrP4CFMsBm10CC3iVwadtqjhYjslprLoj
bkTZV3NmFbwyLTraKy2xOVYq7gyuWFnQysFjlLEiq6hc1AqwQM359oIFhkS3guqtxb8lWhlm
```

```
b6mLZzIWYkZ2lqgAm20kWc6eiJc5ccYq5B98ih3RaZv3k9qXEFoW7SI2eNW8458abUBGJBnb
S9mtRFyvmapGLJnkamGNW3cGSLq5eqrpCkakqE2WlSeZu7e/2pGanHgL8SLAZuI6UYkVSmz4
Q26SHioBjwqIURGHX5ZPGq4LuxAlb4AV+DP/A7QBlas429ORQ/puG2hbnIbqGeiQbD8TitkT
CMGbX2EzBSGQ6FuJ82stA8rLQeVvzFPV3RYRPU60ZZ4iT1w9QDiLaVBasdjHNGTPmoYQ4gGq
ooGLc6CFjSPqibqOUc+3aQN7JF02uxnHCoklWKLaaCtKjDZCTgkZIBA6dOvhH3KcCApwQC59
GA60/l4RnK30lHnna13o+xrJqJcSW9q/cvtmE4b/tDKSJ7Ovk2FaiAf+2sW4NHJrCi7fYLgE
oyma08YpTxjKvEq579UfHu5z/c63N7lze+lHE3WuDqf9YHxZzTV78obKMWgOV5BJlCOh6Oa3
DaN13KosVPlnt06Ab+IKZLFNdGMEYLvTHF+PmFB5yyt8rYJ3RGLKmGJhDOjTrRAnekVVhJHw
xhYhHMD7w2NXm0bSkBLjZ7xnBWNHt4uKlZWz1xlK37YB9kIFCBpG0bg1QsjYXmFfZF/RIinY
BE8Fb5FCwodQ6cAZKZvfeEYy9vRUjhzMy2AG2ogziogkFgOZf0WG2EpNAlcIgLYcYqQtRg0G
ETMAdGmtKd0ewp8HAKnRzIGdGvwjaFu1YJUiTCmje4mjqa1BTJWVq3BuIqN5s462dSY69t/O
6eeHxVUlzdexw8Q9UIeWraqDTQml9Poc/7mipgOSx0KLHPNhYVGUSXAeQ6NHoh30KwMIIG/g
DANBgkqhkiG9w0BAQEFAASCBugwggbkAgEAAoIBgQCc8x0qq2jSuQhAVnTBLJpeYHeUa/x4t
UhbTziOADVcmHwJaupEXAzViCqoQ0Nk44ZmR9sdDPatz/Pil55Ssrv6NAEtGHV5HWkdU0AxA
D5YXQbNCNEDVyAtVKlbYBokr+jS7Ed7In3+5fts5tenP0oCaAD+gi8zbrNffFXowZMzCZByQ
JkvQZjS79tx6YxEHWeTdnmIUpg+ueluRGRw2+hSAFBdPIZXLKDbSR9WrTsS/kEBmrcKZEbEI
yq4iHvivtpi0rgqGjprGvFo2GTvoJ76GOdZEutkU0et+CEbtmexUmySf399o8EufEf293W37
P6aA6ipIf9knBIekPjueWaxOY6AU3aG++htyVL0rlgcxwLKFEqXIM6WvTePG1mkcVNsQSkFr
+Bn/+vYVBJYx1EDTqNj0/fZY1MasYDuWXImx83iCbmBIGzZ7BCAgFL9rc8riKAw5IRVQWXCE
8x4UbQb9L0gq1MnqxVsn5g5MjpG8MCAwEAAQKCAYAMQF7Wr0CxSk5Rlce3y8wh19sdxxXGZz
Miy/SFnF1ayz0x3Jp8fmIy41YCta8NSt5GUULSaeoLniY5WhqHj0b+YsOnI7fWN3lQF7XBqn
WXkjnVexdNvaXxkLoToiHVGuiZJmaO2LduVI3wbbVunpGGNI/EGwdyNvVh9V1QNcqDlMu/KR
UdpW8EVqlFPOgWtrDlVIprtzIGAd8t6cE2Nuq6RybLnTQWml47UxVS+WMy7VphTkfbIM7BWr
E9+BQRok5EmmM2mFEp91iSUXAlRtN7+5MQYhxpXiR3fm+FlTyR9WcnGtYADxZUM2H/+9FLV/
4qXLqDMoVt4D3mf3oMH6FE6y/IN/IqC3NxH1SVYlU7mgsYkSh7XzO0i1rH9oEW4+mc61raj9
KrHkzjKhS02fd9o2oOTX6z+A31+F3h6LukwDAhl4ZsMdHtwF5bzMeRSPhnb3G+Tqo3WXrUfh
oWkTumn+5Uyb4y2ucXlEIjJVwSMECgcEAytJyAIePKdkuTi66UOl3v+cil3iQfeOPR/XhclI
kpqkt+K7HC5tQR1eckfcoiRg7uvOPSGCm1x8EYNqf6cFreofNvhcCY74hyeyuubKeIwcqS11
4vrq0zeVkMznpXhPQaxDr/MPmBF1F/cO+iXEESZ+UQ7C8ou1XSGk2GhIDMMgEJ4xHi+40TJX
rmEuDx2sZuq61OKJ/is7YZRFB+EczXexLKAou+w3vn0rIwivQY2VoP6c09IeNbOhAoHBAMYZ
xKjUCnu81BQ4gm/Z9F1L67ISve2x8mqzvxeveYk79s8KsSyhfNpyW5iYY2mwenSoNfX6mFCI
npZR12po9eNpY9/47fqS5Ds3guoGjve+FZcPx3bEpelCrLPvHKwefz/QdrlcTHSQ84xo3Em3
S8sZSA7n7L4YgrkkcDKAFka7sTgmgE8srwnunYx9fApltvu1B5sY8P0u0Kce3spL5yTywo60
6Tcll0iUlt8KgMYAH55i4hpU4wKBwQCT/Se3oTHhvBkgbNH/tcptmK3RzePIJ2FlhVBBhR/W
RKtJd+W+X04dMB0LGEdYePkkOREEI5m4/2XHcd4axsmwdyS1Hb7dnl6zImJ/FohyUIkYwoy1
xsSkM0uAYTIZNeXuKUO3hAzyhEhMmCYPzQnu2mRpW6zYSMtWB4a1UQ1I67i0TpPp0x4P/5kD
qmb1+/yjDVUnhNIBK1oTrJrxGKqvTnmsNnn/LzT3iq6fly6PR+/f+hYWkk2r+ECgcAvl8H8b
dbC7ZXCGeV0e9eShofkq04ICiVkbzSsuxKm3son6bixMUbusGTDH3Fa1fRFjJgSNZdcWblv2
+Y+AWN5tDszdvmtppbAh1Cjatn0OS20sizIouv9Q3B8dijcTSNjMHEljUAtDRoGFLc/2pnRM
ZkTQ7MD5J2bX165Mz6m1bb4938CBokzCKm/M/KWCOPSSya9o1taaGHmlKsPzPaGUPBfGovID
lWPMWuhwEM5zpPKGJWbn+MCgcEAvXFLGhtsMAJX2PKXSjEnq7NsSNGtI0IJgp/pqu2fbDZf3
okTjcE+SB/TEurfpeM/XGnlxGOz0AnqjrYFy4Ha9borActrMODELIziGVbkGoQCsYCZZqGLb
2pwdQqyJ9WCTRA+4nUvdPWKu0PxyCvbQ51s4/0V+ffpCwjc9WMg0SAkn1C1sWu7yWWRR6bdg
J2yh97mTCfBQXHBeZ3hczweJYx90n1T94G/6ixkCd9ZllX+Oz6a2lBHZ7c
-----END PRIVATE KEY-----
```

B.2.9. id-Kyber512-RSA

TODO

B.2.10. id-Kyber512-ECDH-P256

TODO

B.2.11. id-Kyber512-x25519

TODO

Appendix C. ASN.1 Module

```
<CODE STARTS>

Composite-Keys-2022

DEFINITIONS IMPLICIT TAGS ::= BEGIN

EXPORTS ALL;

IMPORTS
  PUBLIC-KEY, SIGNATURE-ALGORITHM, ParamOptions, AlgorithmIdentifier{}
    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
--

der OBJECT IDENTIFIER ::=
  {joint-iso-itu-t asn1(1) ber-derived(2) distinguished-encoding(1)}

-- To be replaced by IANA
id-composite-key OBJECT IDENTIFIER ::= {
    joint-iso-itu-t(2) country(16) us(840) organization(1) entrust(114027)
    Algorithm(80) Composite(4) CompositeKey(1)



--  COMPOSITE-KEY-ALGORITHM
--
--  Describes the basic properties of a composite key algorithm
--
--  &id - contains the OID identifying the composite algorithm
--  &Params - if present, contains the type for the algorithm
--               parameters; if absent, implies no parameters
--  &paramPresence - parameter presence requirement
--
```

```
-- }

COMPOSITE-KEY-ALGORITHM ::= CLASS {
    &id             OBJECT IDENTIFIER UNIQUE,
    &Params         OPTIONAL,
    &paramPresence  ParamOptions DEFAULT absent
} WITH SYNTAX {
    IDENTIFIER &id
    [PARAMS [TYPE &Params] ARE &paramPresence ]
}


CompositeAlgorithmIdentifier ::= AlgorithmIdentifier{COMPOSITE-KEY-ALGORITHM, {CompositeAlgor

CompositeAlgorithmSet COMPOSITE-KEY-ALGORITHM ::= {
  CompositeAlgorithms, ...
}

--
-- Public Key
--

pk-Composite PUBLIC-KEY ::= {
    IDENTIFIER id-composite-key
    KEY CompositePublicKey
    PARAMS TYPE CompositeAlgorithmIdentifier ARE optional
    PRIVATE-KEY CompositePrivateKey
}

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

CompositePublicKeyOs ::= OCTET STRING (CONTAINING CompositePublicKey ENCODED BY der)

CompositePublicKeyBs ::= BIT STRING (CONTAINING CompositePublicKey ENCODED BY der)

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


-- pk-explicitComposite - Composite public key information object

pk-explicitComposite{OBJECT IDENTIFIER:id, PUBLIC-KEY:firstPublicKey,
  FirstPublicKeyType, PUBLIC-KEY:secondPublicKey, SecondPublicKeyType}
  PUBLIC-KEY ::= {
    IDENTIFIER id
    KEY ExplicitCompositePublicKey{firstPublicKey, FirstPublicKeyType,
      secondPublicKey, SecondPublicKeyType}
    PARAMS ARE absent
}

-- The following ASN.1 object class then automatically generates the
```

```
-- public key structure from the types defined in pk-explicitComposite.

-- ExplicitCompositePublicKey - The data structure for a composite
-- public key sec-composite-pub-keys and SecondPublicKeyType are needed
-- because PUBLIC-KEY contains a set of public key types, not a single
-- type.
-- TODO The parameters should be optional only if they are marked
-- optional in the PUBLIC-KEY


ExplicitCompositePublicKey{PUBLIC-KEY:firstPublicKey, FirstPublicKeyType,
  PUBLIC-KEY:secondPublicKey, SecondPublicKeyType} ::= SEQUENCE {
    firstPublicKey SEQUENCE {
        params firstPublicKey.&Params OPTIONAL,
        publicKey FirstPublicKeyType
    },
    secondPublicKey SEQUENCE {
        params secondPublicKey.&Params OPTIONAL,
        publicKey SecondPublicKeyType
    }
}

END

<CODE ENDS>
```

Appendix D. Intellectual Property Considerations

The following IPR Disclosure relates to this draft:

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

Appendix E. 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), Klaus-Dieter Wirth (D-Trust), 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].

E.1. Making contributions

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

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

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

Jan Klaussner
D-Trust GmbH
Kommandantenstr. 15
10969 Berlin
Germany

Email: jan.klaussner@d-trust.net