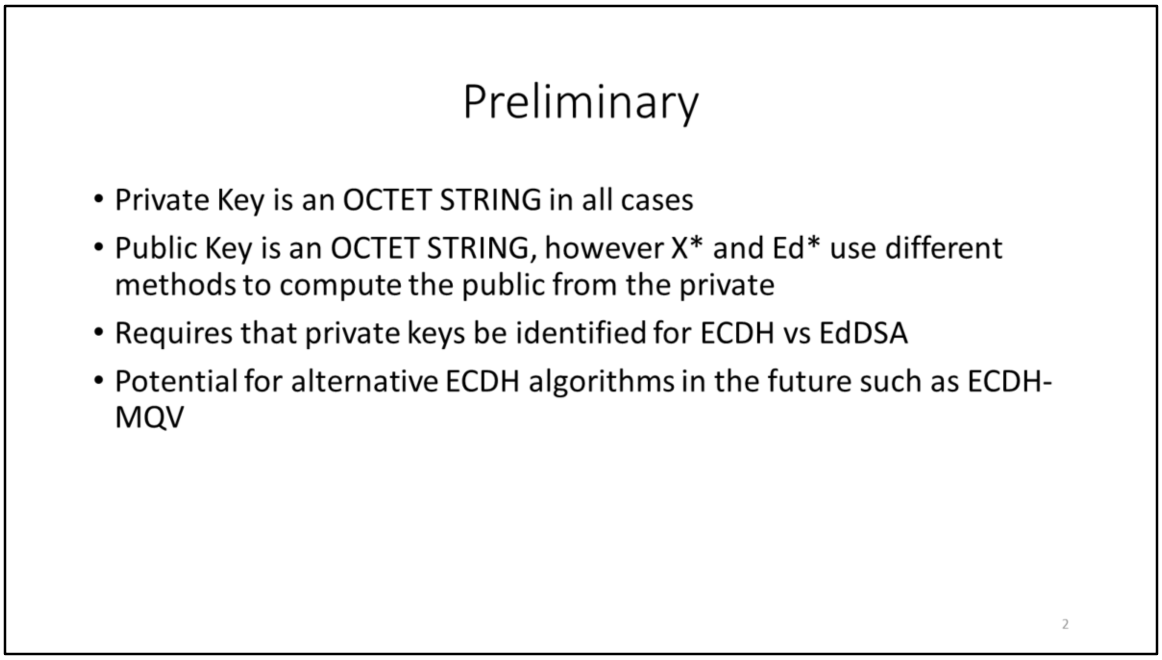
CURDLE

Jim Schaad
August Cellars

IETF 96 - Berlin



Preliminary

- Private Key is an OCTET STRING in all cases
- Public Key is an OCTET STRING, however X* and Ed* use different methods to compute the public from the private
- Requires that private keys be identified for ECDH vs EdDSA
- Potential for alternative ECDH algorithms in the future such as ECDH-MQV

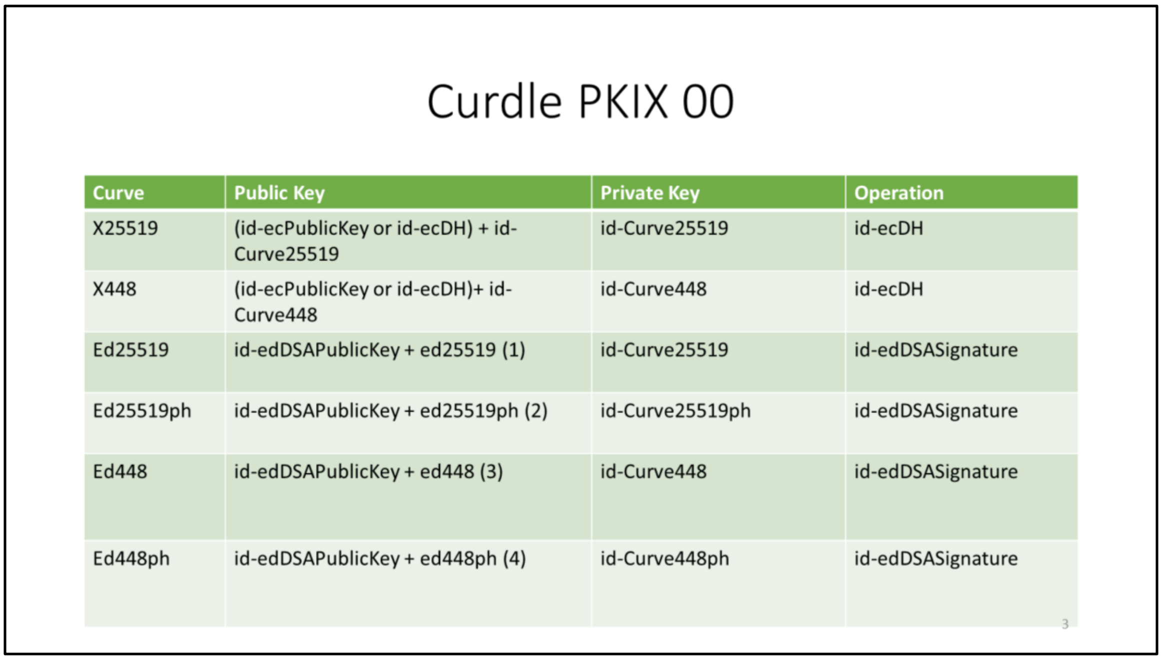
2

Private key is a byte string of either 32-bytes or 56-bytes in both cases
Public key is different for X* and Ed* in the following methods:

1. X* is just the X coordinate while Ed* encodes both the X and the Y coordinates – although only a sign bit for the X coordinate
2. X* computes public by [private]Base while Ed* uses SHA-512 or SHA256 as part of the input process

May want to allow for ECDH methods other than straight ECDH, need to make sure that we understand (and document) what should happen as new ECDH algorithms are adopted (when and if).

Note: known problems with mqv make it unlikely to be used.
EC-MQV is part of Suite B (still true?)

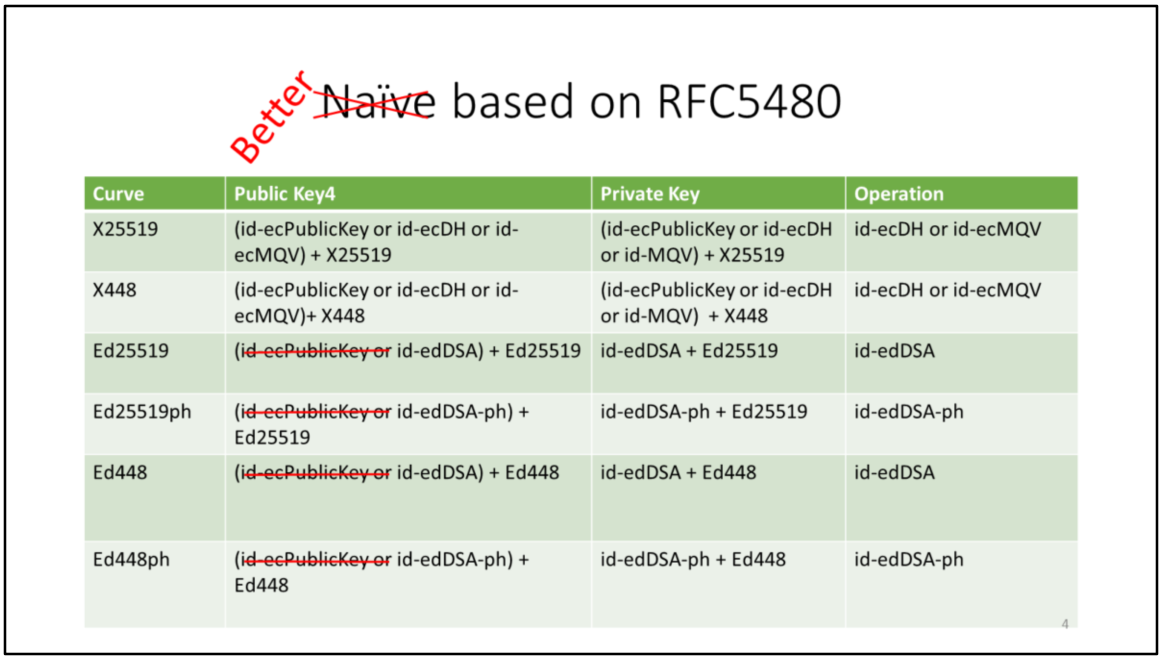


Curdle PKIX 00

Curve	Public Key	Private Key	Operation
X25519	(id-ecPublicKey or id-ecDH) + id-Curve25519	id-Curve25519	id-ecDH
X448	(id-ecPublicKey or id-ecDH)+ id-Curve448	id-Curve448	id-ecDH
Ed25519	id-edDSAPublicKey + ed25519 (1)	id-Curve25519	id-edDSASignature
Ed25519ph	id-edDSAPublicKey + ed25519ph (2)	id-Curve25519ph	id-edDSASignature
Ed448	id-edDSAPublicKey + ed448 (3)	id-Curve448	id-edDSASignature
Ed448ph	id-edDSAPublicKey + ed448ph (4)	id-Curve448ph	id-edDSASignature

3

Nikos – 5/17 – Use OID not enumeration



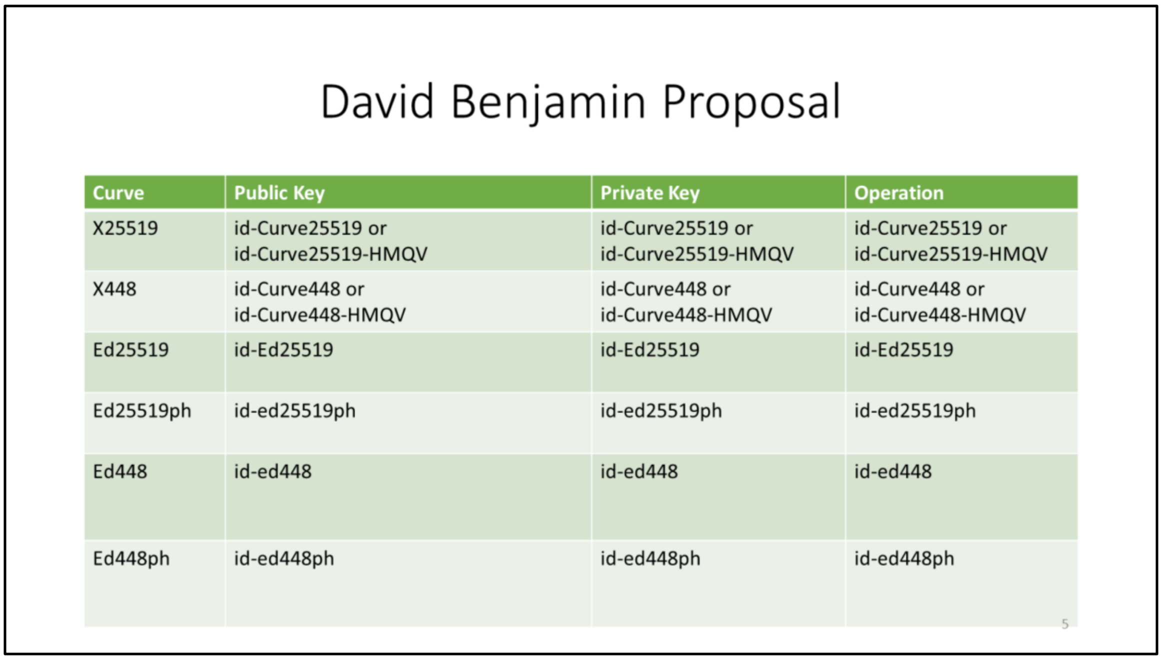
Better ~~Naïve~~ based on RFC5480

Curve	Public Key4	Private Key	Operation
X25519	(id-ecPublicKey or id-ecDH or id-ecMQV) + X25519	(id-ecPublicKey or id-ecDH or id-MQV) + X25519	id-ecDH or id-ecMQV
X448	(id-ecPublicKey or id-ecDH or id-ecMQV)+ X448	(id-ecPublicKey or id-ecDH or id-MQV) + X448	id-ecDH or id-ecMQV
Ed25519	(~~id-ecPublicKey or~~ id-edDSA) + Ed25519	id-edDSA + Ed25519	id-edDSA
Ed25519ph	(~~id-ecPublicKey or~~ id-edDSA-ph) + Ed25519	id-edDSA-ph + Ed25519	id-edDSA-ph
Ed448	(~~id-ecPublicKey or~~ id-edDSA) + Ed448	id-edDSA + Ed448	id-edDSA
Ed448ph	(~~id-ecPublicKey or~~ id-edDSA-ph) + Ed448	id-edDSA-ph + Ed448	id-edDSA-ph

4

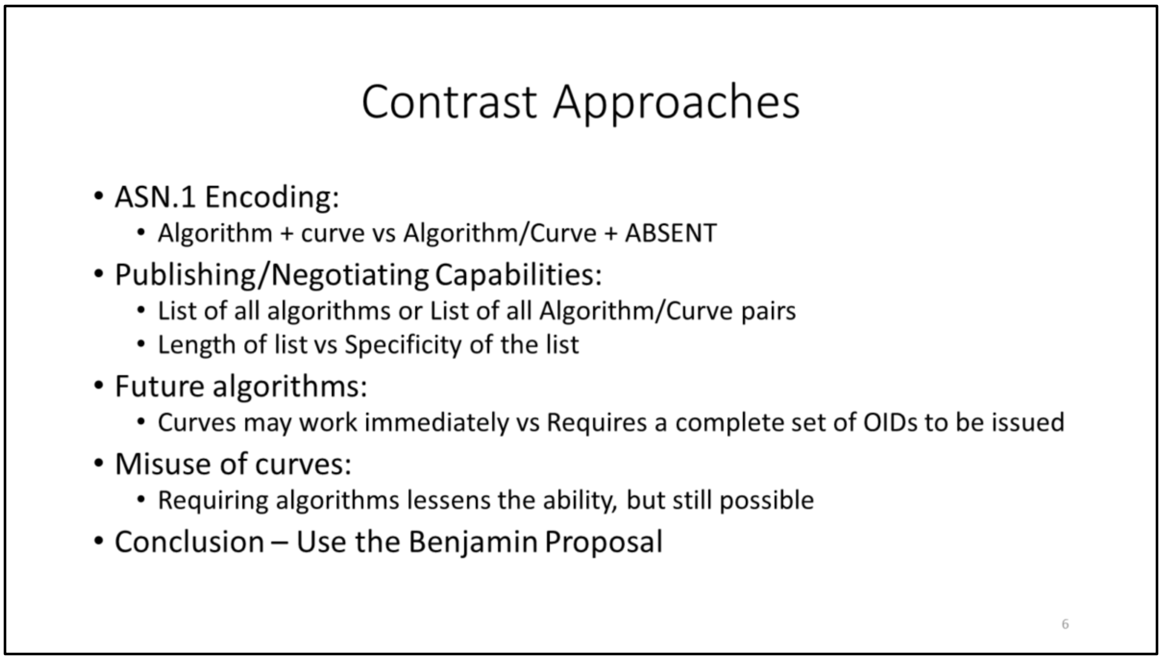
Since public and private keys are not totally matched up, need to get rid of the generic id-ecPublicKey for signatures

If we want to change things to say that cannot use a single key for multiple agorithms on the ECDH side – can kill id-ecPublicKey in those two rows as well.



David Benjamin Proposal

Curve	Public Key	Private Key	Operation
X25519	id-Curve25519 or id-Curve25519-HMQV	id-Curve25519 or id-Curve25519-HMQV	id-Curve25519 or id-Curve25519-HMQV
X448	id-Curve448 or id-Curve448-HMQV	id-Curve448 or id-Curve448-HMQV	id-Curve448 or id-Curve448-HMQV
Ed25519	id-Ed25519	id-Ed25519	id-Ed25519
Ed25519ph	id-ed25519ph	id-ed25519ph	id-ed25519ph
Ed448	id-ed448	id-ed448	id-ed448
Ed448ph	id-ed448ph	id-ed448ph	id-ed448ph



Contrast Approaches

- ASN.1 Encoding:
  - Algorithm + curve vs Algorithm/Curve + ABSENT
- Publishing/Negotiating Capabilities:
  - List of all algorithms or List of all Algorithm/Curve pairs
  - Length of list vs Specificity of the list
- Future algorithms:
  - Curves may work immediately vs Requires a complete set of OIDs to be issued
- Misuse of curves:
  - Requiring algorithms lessens the ability, but still possible
- Conclusion – Use the Benjamin Proposal

6

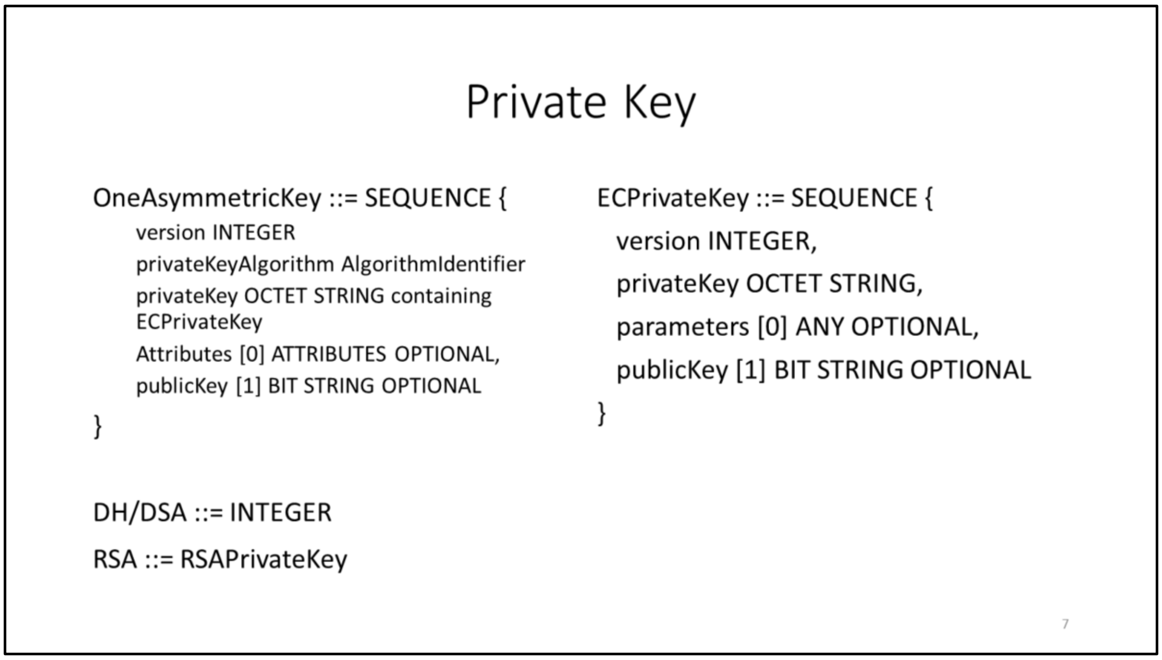
I am going to ignore the current document as there does not seem to be a great deal of support for it.
Based on the ASN.1 encoding, not a great deal to recommend either one. Slightly favor the second approach due to size

Publishing Capabilities:
- Shorter list if just algorithms
- More likely to affect signatures than ECDH as still need to provide a key
- May assume that algorithms in the list are still ok if list is long enough – shorten to favorites rather than complete support
- Possible to produce a new OID which is just for negotiation if it looks like it really needs to exist.
- LDAP searches for algorithm are more complicated as a larger list of algorithms needs to be done.

Future Algorithms:
- Case of how MQV would be added to the list of algorithms
- Does not work immediately w/o generic curve identifier. Thus does not affect how signature curves are setup.

Curve Misuse:
- Listing algorithms make misuse more blatant. More likely for id-ecPublicKey than edDSA keys. Stupid people can still do stupid things

Memory of somebody saying that knowing just the alg w/o the curve is a problem in some circumstances. – looking at signatures need to go and find the key to know if you can process it. I am not sure if I feel that this is a real problem as you are going to have to spend a lot of time dealing with the certificate before you can use it. Not sure how many instances of – I can’t do this curve – exist in real life.



Private Key

```
OneAsymmetricKey ::= SEQUENCE {
    version INTEGER
    privateKeyAlgorithm AlgorithmIdentifier
    privateKey OCTET STRING containing
    ECPrivateKey
    Attributes [0] ATTRIBUTES OPTIONAL,
    publicKey [1] BIT STRING OPTIONAL
}
```

```
ECPrivateKey ::= SEQUENCE {
  version INTEGER,
  privateKey OCTET STRING,
  parameters [0] ANY OPTIONAL,
  publicKey [1] BIT STRING OPTIONAL
}
```

```
DH/DSA ::= INTEGER
RSA ::= RSAPrivateKey
```

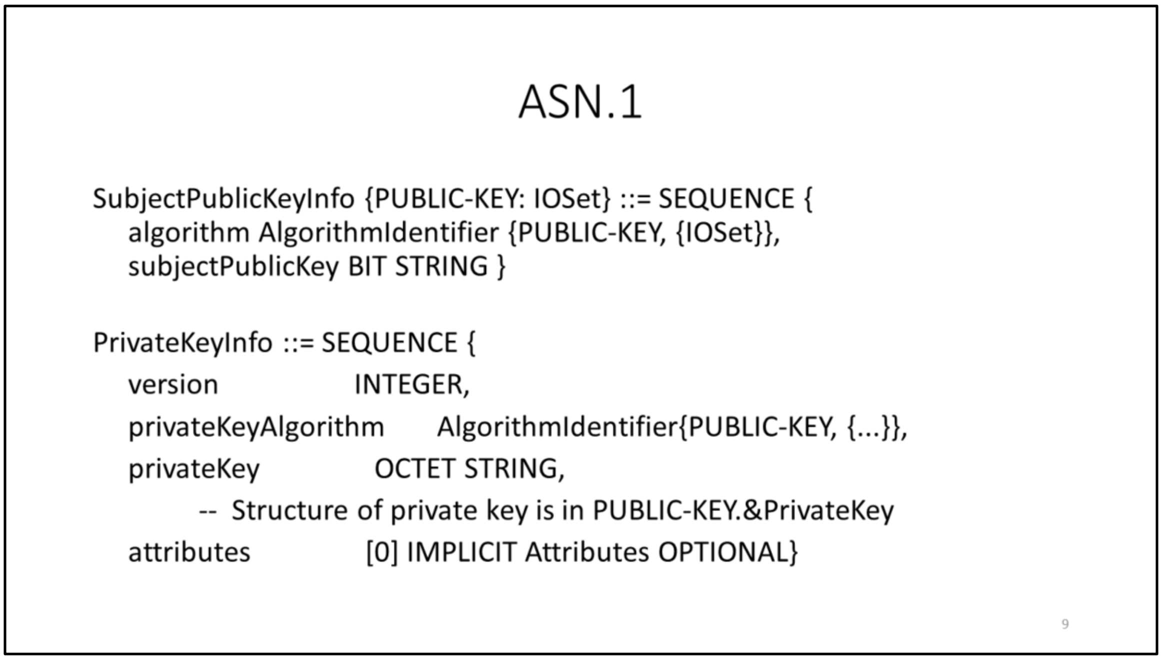
7

Makes more sense to do what DH does and say – ECPrivateKey is an OCTET STRING? And just use the OneAsymmetricKey structure as a wrapper (almost the same as PKCS#8

RFC 5958

Recommendations on sending a public key with the private key? Should we make one.

Discussions



ASN.1

```
SubjectPublicKeyInfo {PUBLIC-KEY: IOSet} ::= SEQUENCE {
    algorithm AlgorithmIdentifier {PUBLIC-KEY, {IOSet}},
    subjectPublicKey BIT STRING }

PrivateKeyInfo ::= SEQUENCE {
    version                 INTEGER,
    privateKeyAlgorithm     AlgorithmIdentifier{PUBLIC-KEY, {...}},
    privateKey              OCTET STRING,
            -- Structure of private key is in PUBLIC-KEY.&PrivateKey
    attributes              [0] IMPLICIT Attributes OPTIONAL}
```

9