New algorithm IDs for the CFRG algorithms in OpenPGP

Simplifying the use of Ed25519, Ed448, X25519, and X448
A bit of history
A bit of history

- June 2012: RFC 6637 - Elliptic Curve Cryptography (ECC) in OpenPGP
A bit of history

● June 2012: RFC 6637 - Elliptic Curve Cryptography (ECC) in OpenPGP

● March 2016: draft-koch-openpgp-rfc4880bis-02 added EdDSA and Ed25519, using OID 1.3.6.1.4.1.11591.15.1 ("ed25519", under private enterprises)
A bit of history

- June 2012: RFC 6637 - Elliptic Curve Cryptography (ECC) in OpenPGP

- March 2016: draft-koch-openpgp-rfc4880bis-02 added EdDSA and Ed25519, using OID 1.3.6.1.4.1.11591.15.1 (“ed25519”, under private enterprises)

- June 2017: draft-ietf-openpgp-rfc4880bis-02 added Curve25519 over ECDH, using OID 1.3.6.1.4.1.3029.1.5.1 (“curvey25519”, now fixed)
A bit of history

- June 2012: RFC 6637 - Elliptic Curve Cryptography (ECC) in OpenPGP

- March 2016: draft-koch-openpgp-rfc4880bis-02 added EdDSA and Ed25519, using OID 1.3.6.1.4.1.11591.15.1 ("ed25519", under private enterprises)

- June 2017: draft-ietf-openpgp-rfc4880bis-02 added Curve25519 over ECDH, using OID 1.3.6.1.4.1.3029.1.5.1 ("curve25519", now fixed)

- August 2018: RFC 8410 - Algorithm Identifiers for Ed25519, Ed448, X25519, and X448 for Use in the Internet X.509 Public Key Infrastructure
More recent history
More recent history

- Last year: discussion in the DT about using new OIDs or new algorithm IDs
More recent history

- Last year: discussion in the DT about using new OIDs or new algorithm IDs
- Decided against it due to deployed base of Curve25519 keys
More recent history

- Last year: discussion in the DT about using new OIDs or new algorithm IDs
- Decided against it due to deployed base of Curve25519 keys
- February: new discussion about the OIDs on the WG mailing list
More recent history

- Last year: discussion in the DT about using new OIDs or new algorithm IDs
- Decided against it due to deployed base of Curve25519 keys

- February: new discussion about the OIDs on the WG mailing list
- Proposal to switch to the X.509 OIDs
More recent history

- Last year: discussion in the DT about using new OIDs or new algorithm IDs
- Decided against it due to deployed base of Curve25519 keys

- February: new discussion about the OIDs on the WG mailing list
- Proposal to switch to the X.509 OIDs
- Proposal to switch to new algorithm IDs
Ed25519 and Ed448
Ed25519 and Ed448

- Hardcoded lengths for keys and signatures
Ed25519 and Ed448

- Hardcoded lengths for keys and signatures
- No MPIs
X25519 and X448
X25519 and X448

- Hardcoded lengths for keys and ephemeral points
X25519 and X448

- Hardcoded lengths for keys and ephemeral points
- No MPIs
X25519 and X448

- Hardcoded lengths for keys and ephemeral points
- No MPIs
- No padding
X25519 and X448

- Hardcoded lengths for keys and ephemeral points
- No MPIs
- No padding
- No two-octet checksum
X25519 and X448

- Hardcoded lengths for keys and ephemeral points
- No MPIs
- No padding
- No two-octet checksum
- HKDF over the shared key
X25519 and X448

- Hardcoded lengths for keys and ephemeral points
- No MPIs
- No padding
- No two-octet checksum
- HKDF over the shared key
- Hardcoded info parameter
X25519 and X448

- Hardcoded lengths for keys and ephemeral points
- No MPIs

- No padding
- No two-octet checksum

- HKDF over the shared key
- Hardcoded info parameter

- Derived key used to encrypt the session key using AES-KW
What about v3 PKESK?
What about v3 PKESK?

- In a v3 PKESK, the symmetric algorithm ID was prepended to the session key.
What about v3 PKESK?

- In a v3 PKESK, the symmetric algorithm ID was prepended to the session key
- AES-KW can only encrypt a multiple of 8 octets
What about v3 PKESK?

- In a v3 PKESK, the symmetric algorithm ID was prepended to the session key
- AES-KW can only encrypt a multiple of 8 octets
- Add padding?
What about v3 PKESK?

● In a v3 PKESK, the symmetric algorithm ID was prepended to the session key
● AES-KW can only encrypt a multiple of 8 octets

● Add padding?
● Leave it unencrypted instead
What about v3 PKESK?

- In a v3 PKESK, the symmetric algorithm ID was prepended to the session key
- AES-KW can only encrypt a multiple of 8 octets
- Add padding?
- Leave it unencrypted instead
- But: concern of cross-algorithm attacks
What about v3 PKESK?

- In a v3 PKESK, the symmetric algorithm ID was prepended to the session key
- AES-KW can only encrypt a multiple of 8 octets

- Add padding?
- Leave it unencrypted instead

- But: concern of cross-algorithm attacks
- Mandate using AES?
ESK is not bound to recipient fingerprint
ESK is not bound to recipient fingerprint

- Recipient fingerprint is not included in the KDF, unlike ECDH
ESK is not bound to recipient fingerprint

- Recipient fingerprint is not included in the KDF, unlike ECDH
- No guarantee that the message was originally encrypted for you
ESK is not bound to recipient fingerprint

- Recipient fingerprint is not included in the KDF, unlike ECDH
- No guarantee that the message was originally encrypted for you
- Need the Intended Recipient Fingerprint signature subpacket for that
ESK is not bound to recipient fingerprint

- Recipient fingerprint is not included in the KDF, unlike ECDH
- No guarantee that the message was originally encrypted for you
- Need the Intended Recipient Fingerprint signature subpacket for that

- This also makes automatic forwarding easier
ESK is not bound to recipient fingerprint

- Recipient fingerprint is not included in the KDF, unlike ECDH
- No guarantee that the message was originally encrypted for you
- Need the Intended Recipient Fingerprint signature subpacket for that

- This also makes automatic forwarding easier
- More about that later
Questions?