Key-Update Mechanism for SDLSP

Andreas Hülsing Tanja Lange **Fiona Weber**

TU/e

27. May 2024

Author list in alphabetical order, see https://www.ams.org/profession/leaders/culture/CultureStatement04.pdf.

Section 1

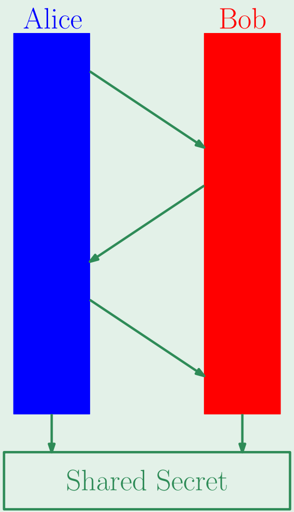
Authenticated Key Exchange

Motivation

- SDLSP secures communication with symmetric keys.
- These *can* be replaced, but the update uses only symmetric cryptography.
    - Cannot recover from corruption!
    - The total number of keys grows quadratically with the number of parties.
    - The number of keys that a party has to know up-front grows linearly.
- Future mega-constellations may massively increase the number of communicating parties.

Authenticated Key Exchange – In General

- Two parties, each with a long-term key-pair for authentication
- At least one party usually generates an ephemeral key-pair
  - Not used outside the exchange, secret-key disposed after exchange.
- The final output of an AKE is a shared secret that only the involved parties know.



Authenticated Key Exchange – In Our Use-Case

- Mission-Control and the Satellite both have a key-pair to authenticate themselves.
- They may have a previous shared secret. (The previous symmetric key)
- AKE computes a new shared secret that is secure even if the old one is leaked.
- Both parties can be certain of the identity of their peer.
- Can be run independently of a messaging-phase.

Advantages

- Total keys only scale *linearly* with the number of parties.
- Usable with a Public-Key-Infrastructure (PKI) – No need to preload all keys.
- Possible to recover from corruption.

Security-Goals

Confidentiality

Attacker does not learn information about resulting key.

- Forward-Secrecy: Even if he later corrupts a party.
- Post-Compromise-Secrecy: Even if he had corrupted the party before.
- Long-Term Security: Deal with “store-now, decrypt-later”-attacks.

Authenticity

Attacker cannot impersonate a different party.

- Prevent replay-attacks (common vulnerability).
- Good news: Attacks inherently have to be performed “live”.

Hybrid Security

- Use two schemes in case one is broken
- Typically EC-schemes, e.g. Hashed Diffie-Hellman using X25519 and ECDSA.
- Can be done on protocol or primitive-level
  - primitive-level is generally simpler
  - it also results in an primitive-agnostic protocol ⇒ More options for implementers
- Fallback does not necessarily have to be pre-quantum!
- Combination trivial for Signatures.
- Less trivial for KEMs, but Hashing shared secrets and ciphertexts works.

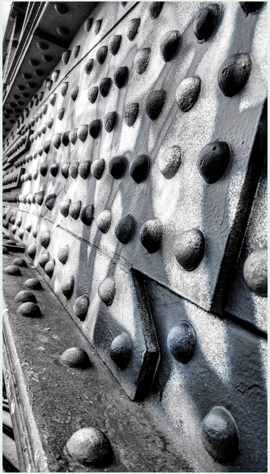


Figure 1: CC-BY-SA 4.0, Michael Musto

Updating long-term keys

- Long-term keys may also get corrupted → should be updatable as well.
- Our protocol contains a mechanism for that.

Unauthenticated Satellites

- Satellites are on publicly known orbits
- Communication-channels are physically narrow
- Physical location could be used for Authentication

- Potential for significant bandwidth-savings.
- Requires that Mission-Control can trust the ground-stations!

⇒ An interesting option that **requires** careful analysis

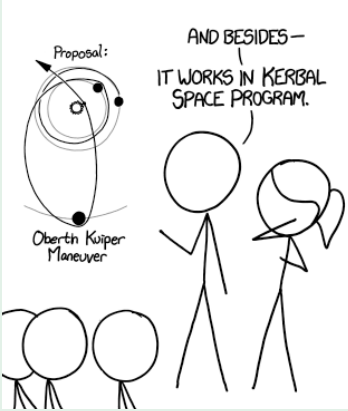


Figure 2: CC-BY-NC 2.5 Randall Munroe, xkcd.com/1244

Section 2

Possible Approaches

Signatures + KEM

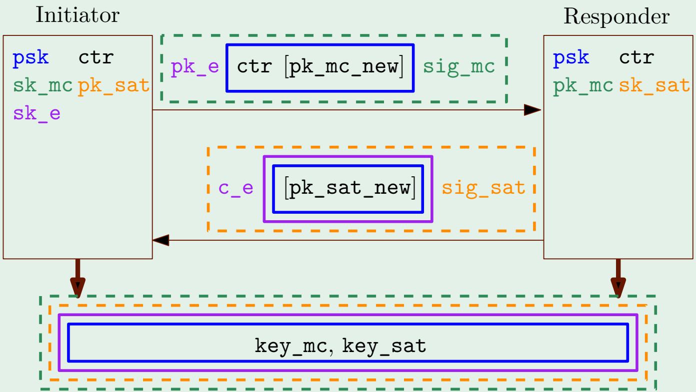


Figure 3: Signatures+KEM: The traditional Way.

- Requires replay-protection! (ctr)
- 1 Roundtrip
- Key-confirmation sensible, but not required.
- long-term-key-updates required if signature-scheme is stateful.
- Stateful scheme would enable few- and one-time signatures.

Triple-KEM and Dual-KEM

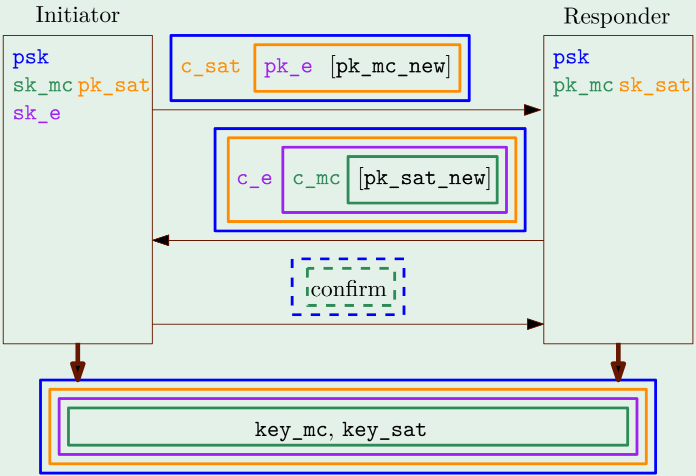


Figure 4: Triple-KEM: The more modern way.

- Usually more efficient (KEMs instead of signatures).
- Essentially invulnerable to replay-attacks.
- Option to mix KEMs.
- Dropping c_sat, pk_sat, pk_sat_new and sk_sat gives **Dual-KEM**, which does not authenticate the receiver.

Considered KEMs

- The “Obvious” Choice: Kyber (NIST: ML-KEM)
- Ten times larger: Frodo
- Worth a look for special use-cases: Classic McEliece
- Not Size-Competitive with Kyber: BIKE and HQC
- Similar to Kyber, but lost PQC: Saber, NTRU, NTRU prime
- Broken: SIKE

Section 3

Our Recommendations

Our Recommendations

Our primary recommendation for general use is:

- **Triple-KEM**, using **Kyber** (and X25519) for all three KEMs

If satellite-authenticity is a given and the bandwidth-savings are important:

- **Dual-KEM**, using **Kyber** (and X25519) for both KEMs

Triple-KEM with Kyber

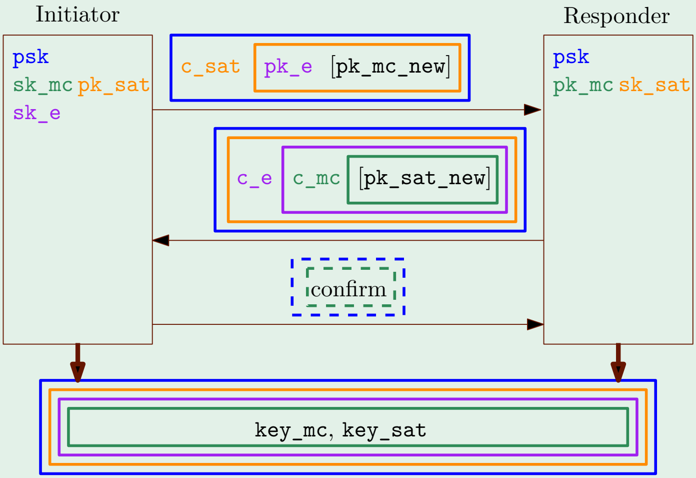


Figure 5: Triple-KEM

Packet sizes in bytes at different security-levels:

- Level 1: 1664, 1632, 16
- Level 3: 2368, 2272, 16
- Level 5: 3232, 3232, 16

With long-term-key updates:

- Level 1: 2496, 2480, 16
- Level 3: 3584, 3504, 16
- Level 5: 4832, 4848, 16

Security Analysis

We analyzed the protocol in a custom eCK-NEC model (= eCK, No Ephemeral Corruption)

- Simplified version of established eCK-model
- Assumes ephemeral randomness cannot be corrupted.
- Provides strong Confidentiality and Authenticity guarantees.

eCK-NEC

Security is usually defined via a "Game" in which an adversary tries to reach a winning-condition.

- n_i initiators and n_r responders run up to n_{s_i}/n_{s_r} initiator/responder-sessions each
- Adversary controls parties actions and the network
- Adversary can corrupt long-term keys and session-keys
- Winning conditions forbid trivial attacks
- Adversary wins
  - if he is able to distinguish an honestly generated key from randomness, or
  - if he is able to impersonate a party without corrupting its long-term-key.

Security

Proven for Triple-KEM in eCK-NEC-model under reasonable assumptions:

- **Honestly generated keys are indistinguishable from randomness.** (Confidentiality)
- **A party cannot be impersonated, as long as its long-term public key remains uncorrupted.** (Authenticity)

Conjectured:

- Honestly generated keys remain confidential if the pre-shared key remains uncorrupted.
- Honestly generated keys remain confidential as long as one party's long-term key and the peer's ephemeral randomness remain uncorrupted.
- As long as a connection remains confidential (see above), no passive attacker can learn more about a new long-term public-key than can be extracted from ciphertexts for that public key. (Identity Hiding)

The same holds for **Dual-KEM**, ***if*** *responder-authenticity is guaranteed out-of-band*.

Conclusion

- Enable asymmetric key-updates for better scaling and security.
- Use post-quantum-secure algorithms for long-term security.
- Use an Authenticated Key Exchange (AKE) as Key-Update Mechanism
- Our Recommendation: Triple-KEM with Kyber+X25519
- Proposal builds on Post-Quantum Noise
- Formal Security-analysis in a simpler version of a standard model.

Conclusion

- Enable asymmetric key-updates for better scaling and security.
- Use post-quantum-secure algorithms for long-term security.
- Use an Authenticated Key Exchange (AKE) as Key-Update Mechanism
- Our Recommendation: Triple-KEM with Kyber+X25519
- Proposal builds on Post-Quantum Noise
- Formal Security-analysis in a simpler version of a standard model.

Questions?

Section 4

Appendix

KEMs – Sizes and Failure-rates

Scheme	SK	PK	CT	δ
X25519	32	32	32	0
Kyber-512	1632	800	768	2^{-139}
Kyber-768	2400	1184	1088	2^{-164}
Kyber-1024	3168	1568	1568	2^{-174}
mceliece348864	6492	261120	96	0
mceliece460896	13608	524160	156	0
mceliece6688128	13932	1044992	208	0
mceliece6960119	13948	1047319	194	0
mceliece8192128	14120	1357824	208	0
FrodoKEM-640	19888	9616	9720	$2^{-138.7}$
FrodoKEM-976	31296	15632	15744	$2^{-199.6}$
FrodoKEM-1344	43088	21520	21632	$2^{-252.5}$

Signatures – Sizes

Scheme	SK	PK	Sig
Dilithium2	2544	1312	2420
Dilithium3	4016	1952	3293
Dilithium5	4880	2592	4595
Falcon-512	1281	897	666
Falcon-1024	2305	1793	1280
ECDSA	32	32	64

Triple-KEM – Packet Sizes

Scheme	Packet 1	Packet 2	Packet 3
TK(Kyber512+X25519)	1664	1632	16
TKU(Kyber512+X25519)	2496	2480	16
TK(Kyber768+X25519)	2368	2272	16
TKU(Kyber768+X25519)	3584	3504	16
TK(Kyber1024+X25519)	3232	3232	16
TKU(Kyber1024+X25519)	4832	4848	16

Sign + KEM – Packet Sizes

Scheme	Packet 1	Packet 2	Packet 3
SK(Kyber512+X25519+Dilithium+ECDSA)	3348	3300	16
SKU(Kyber512+X25519+Dilithium+ECDSA)	4692	4644	16
SK(Kyber512+X25519+Falcon+ECDSA)	1594	1546	16
SKU(Kyber512+X25519+Falcon+ECDSA)	2523	2475	16
SK(Kyber512+X25519+XMSS-SHA2_10_256)	3364	3316	16
SKU(Kyber512+X25519+XMSS-SHA2_10_256)	3428	3380	16
SC(Kyber512+X25519,WOTS+(32,16))	3024	2992	16
SC(Kyber768+X25519,WOTS+(32,16))	2408	3312	16
SC(Kyber1024+X25519,WOTS+(32,16))	3792	3792	16
SC(Kyber1024+X25519,WOTS+(64,16))	10032	10032	16

Formal Security Triple-KEM

There is no adversary that can win the eCK-NEC-game against Triple-KEM, with:

$$\mathsf{Adv}_{\mathcal{A},\,\mathsf{3KEM}}^{\mathsf{eCK\text{-}NEC}}\left(1^\lambda\right) \leq \left(\begin{array}{crcl} & 3 & \cdot & \mathsf{Adv}_{\mathcal{A}_1,\,\mathsf{H}}^{\mathsf{coll\text{-}res}}\left(1^\lambda\right) \\ + & n_i \cdot n_{s_i} & \cdot & \mathsf{EKEM}.\delta \\ + & n_i \cdot n_{s_i} \cdot n_r \cdot n_{s_r} \cdot 3 & \cdot & \mathsf{Adv}_{\mathcal{A},\,\mathsf{EKEM}}^{\mathsf{IND\text{-}CCA}}\left(1^\lambda\right) \\ + & n_{s_r} \cdot n_i \cdot n_r \cdot \frac{1}{1-\mathsf{IKEM}\cdot\delta} & \cdot & \mathsf{Adv}_{\mathcal{A}_4,\,\mathsf{IKEM}}^{\mathsf{IND\text{-}CCA}}\left(1^\lambda\right) \\ + & n_{s_i} \cdot n_i \cdot n_r \cdot \frac{1}{1-\mathsf{RKEM}\cdot\delta} & \cdot & \mathsf{Adv}_{\mathcal{A}_4,\,\mathsf{RKEM}}^{\mathsf{IND\text{-}CCA}}\left(1^\lambda\right) \\ + & n_i \cdot n_{s_i} \cdot n_r \cdot n_{s_r} \cdot 3 & \cdot & \mathsf{Adv}_{\mathcal{A},\,\mathsf{NHO}}^{\mathsf{PRHO}}\left(1^\lambda\right) \\ + & \left(n_{s_i} + n_{s_r}\right) \cdot n_i \cdot n_r & \cdot & \mathsf{Adv}_{\mathcal{A}_6,\,\mathsf{AEAD}}^{\mathsf{EUF\text{-}CMA}}\left(1^\lambda\right) \\ + & n_i \cdot n_{s_i} \cdot n_r \cdot n_{s_r} \cdot 2 & \cdot & \mathsf{Adv}_{\mathcal{A},\,\mathsf{KDF}}^{\mathsf{PRF}}\left(1^\lambda\right) \end{array}\right)$$

Formal Security Dual-KEM

There is no adversary that can win the eCK-NEC-game against Dual-KEM, with:

$$
\mathsf{Adv}^{\text{eCK-NEC}}_{\mathcal{A},\,\text{2KEM}}\left(1^\lambda\right) \leq \left(\begin{array}{rrcl}
 & 2 & \cdot & \mathsf{Adv}^{\text{coll-res}}_{\mathcal{A}_1,\,\mathsf{H}}\left(1^\lambda\right) \\
+ & n_i \cdot n_{s_i} & \cdot & \text{EKEM}.\delta \\
+ & n_i \cdot n_{s_i} \cdot n_r \cdot n_{s_r} \cdot 3 & \cdot & \mathsf{Adv}^{\text{IND-CCA}}_{\mathcal{A},\,\text{EKEM}}\left(1^\lambda\right) \\
+ & n_{s_r} \cdot n_i \cdot n_r \cdot \frac{1}{1-\text{IKEM}\cdot\delta} & \cdot & \mathsf{Adv}^{\text{IND-CCA}}_{\mathcal{A}_4,\,\text{IKEM}}\left(1^\lambda\right) \\
+ & n_i \cdot n_{s_i} \cdot n_r \cdot n_{s_r} \cdot 2 & \cdot & \mathsf{Adv}^{\text{PRHO}}_{\mathcal{A},\,\text{NHO}}\left(1^\lambda\right) \\
+ & n_{s_r} \cdot n_i \cdot n_r & \cdot & \mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_6,\,\text{AEAD}}\left(1^\lambda\right) \\
+ & n_i \cdot n_{s_i} \cdot n_r \cdot n_{s_r} \cdot 2 & \cdot & \mathsf{Adv}^{\text{PRF}}_{\mathcal{A},\,\text{KDF}}\left(1^\lambda\right) \\
+ & & & \mathsf{Adv}^{\text{eCK-NEC}_{\text{Case A}}}_{\mathcal{A},\,\text{2KEM}}\left(1^\lambda\right)
\end{array}\right)
$$

Where $\mathsf{Adv}^{\text{eCK-NEC}_{\text{Case A}}}_{\mathcal{A},\,\text{2KEM}}\left(1^\lambda\right)$ Refers to the maximum achievable advantage for the adversary to cause an unpeered, complete initiator-session.