Transitioning to post-quantum: How PQC affects protocols and what we can do today?

Tanja Lange

Eindhoven University of Technology

02 Sep 2021

How does PQC affect protocols?

- Length fields don't fit.

How does PQC affect protocols?

- Length fields don't fit.
  $\Rightarrow$ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to
  PQC one, putting PQC part into the payload.

How does PQC affect protocols?

- Length fields don't fit.
  $\Rightarrow$ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to
  PQC one, putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.

How does PQC affect protocols?

- Length fields don't fit.
  $\Rightarrow$ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to
  PQC one, putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.
  Most experiments don't look so devastating.
- Interface mismatch – KEM instead of key exchange, stateful hash-based signatures fit some applications.

How does PQC affect protocols?

- Length fields don't fit.
  $\Rightarrow$ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to
  PQC one, putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.
  Most experiments don't look so devastating.
- Interface mismatch – KEM instead of key exchange, stateful hash-based signatures fit some applications.
  $\Rightarrow$ Shoehorning PQC into current systems may prioritize weaker systems; redesigning protocols takes time.

How does PQC affect protocols?

- Length fields don't fit.
  $\Rightarrow$ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to
  PQC one, putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.
  Most experiments don't look so devastating.
- Interface mismatch – KEM instead of key exchange, stateful hash-based signatures fit some applications.
  $\Rightarrow$ Shoehorning PQC into current systems may prioritize weaker systems; redesigning protocols takes time.
- Validation and certification schemes are not yet updated.

How does PQC affect protocols?

- Length fields don't fit.
  $\Rightarrow$ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to
  PQC one, putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.
  Most experiments don't look so devastating.
- Interface mismatch – KEM instead of key exchange, stateful hash-based signatures fit some applications.
  $\Rightarrow$ Shoehorning PQC into current systems may prioritize weaker systems; redesigning protocols takes time.
- Validation and certification schemes are not yet updated.
  $\Rightarrow$ Combine pre-and post-quantum schemes,
  certification only applies to pre-quantum scheme.

How does PQC affect protocols?

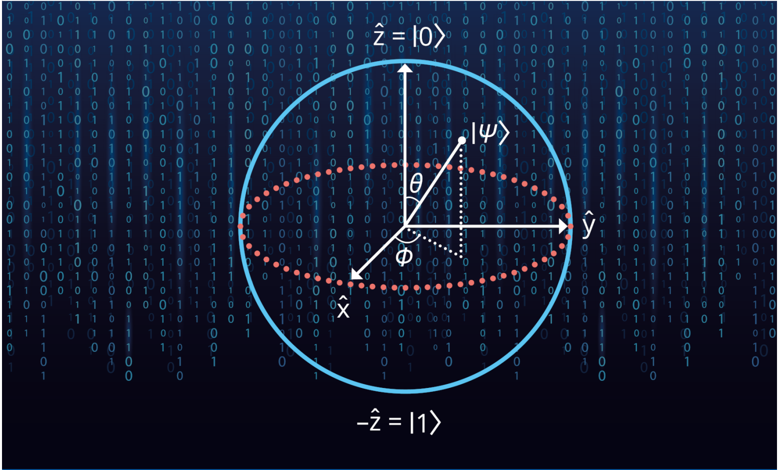
- Length fields don't fit.
  $\Rightarrow$ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to
  PQC one, putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.
  Most experiments don't look so devastating.
- Interface mismatch – KEM instead of key exchange, stateful hash-based signatures fit some applications.
  $\Rightarrow$ Shoehorning PQC into current systems may prioritize weaker systems; redesigning protocols takes time.
- Validation and certification schemes are not yet updated.
  $\Rightarrow$ Combine pre-and post-quantum schemes,
  certification only applies to pre-quantum scheme.
  For such *hybrid* schemes, ensure that as strong as strongest not as weak as weakest.

How does PQC affect protocols?

- Length fields don't fit.
  $\Rightarrow$ Restrict to systems that fit, if any,
  or keep pre-quantum algorithm next to
  PQC one, putting PQC part into the payload.
- Speed, resources.
  Combined schemes take about twice the time.
  Most experiments don't look so devastating.
- Interface mismatch – KEM instead of key exchange, stateful hash-based signatures fit some applications.
  $\Rightarrow$ Shoehorning PQC into current systems may prioritize weaker systems; redesigning protocols takes time.
- Validation and certification schemes are not yet updated.
  $\Rightarrow$ Combine pre-and post-quantum schemes,
  certification only applies to pre-quantum scheme.
  For such *hybrid* schemes, ensure that as strong as strongest not as weak as weakest.
- New security assumptions, new proofs, lots of new code.

Post-Quantum Cryptography: Current state and quantum mitigation

Ward Beullens, Jan-Pieter DAnvers, Andreas Hülsing, Tanja Lange, Lorenz Panny, Cyprien de Saint Guilhem, Nigel P. Smart. Evangelos Rekleitis, Angeliki Aktypi, Athanasios-Vasileios Grammatopoulos.



POST-QUANTUM CRYPTOGRAPHY

Current state and quantum mitigation

ENISA report: Current state and quantum mitigation

Chapters

1. Introduction
2. Families of Post-Quantum Algorithms
3. Security Notions and Generic Transforms
4. NIST Round 3 Finalists
5. Alternate Candidates
6. Quantum Mitigation
   - 6.1 Hybrid schemes
   - 6.2 Protective measures for pre-quantum cryptography

Report available from ENISA's website.

Hybrid schemes

Combine one (or more) pre-quantum schemes with one (or more) post-quantum schemes.

Signatures:
All individual signatures must be valid for the hybrid signature to be valid.

Hybrid schemes

Combine one (or more) pre-quantum schemes with one (or more) post-quantum schemes.

Signatures:

All individual signatures must be valid for the hybrid signature to be valid.

DH / KEM:

Use KDF on concatenation of keys or consume iteratively.

Different options to hybridize

- Execute pre- and post-quantum next to each other.
- Wrap PQC inside pre-quantum (benefit for length fields).
- Wrap pre-quantum inside PQC (limit the attack surface – quantum attacker cannot even break pre-quantum scheme).

Protective measures for pre-quantum cryptography

Aka poor-man's PQC

Premise: Known/slowly changing set of peers.

This fits email, messaging (Signal, etc.), most enterprise setups.

Does not fit web servers.

Protective measures for pre-quantum cryptography

Aka poor-man's PQC

Premise: Known/slowly changing set of peers.
This fits email, messaging (Signal, etc.), most enterprise setups.
Does not fit web servers.

Requires: Adjust protocol to have user keep state per peer.

Protective measures for pre-quantum cryptography

Aka poor-man's PQC

Premise: Known/slowly changing set of peers.
This fits email, messaging (Signal, etc.), most enterprise setups.
Does not fit web servers.

Requires: Adjust protocol to have user keep state per peer.

Option 1: Have fixed secret per peer, include this in KDF.
Secret exchanged out of band, or exchange is not observed.
Provided in WireGuard as option.

Protective measures for pre-quantum cryptography

Aka poor-man's PQC

Premise: Known/slowly changing set of peers.
This fits email, messaging (Signal, etc.), most enterprise setups.
Does not fit web servers.

Requires: Adjust protocol to have user keep state per peer.

Option 1: Have fixed secret per peer, include this in KDF.
Secret exchanged out of band, or exchange is not observed.
Provided in WireGuard as option.

Option 2: Have updatable secret per peer, include this in KDF.
Update per-peer secret with each new public-key operation.
Initial secret exchanged out of band, or exchange is not observed.

More complicated dataflow, e.g., do not overwrite without confirmation that peer can update, but full forward secrecy.

Details worked out in RFC 6189 on ZRTP, see also section 6.2 of the ENISA report.

PQConnect: An Automated Boring Protocol for Quantum-Secure Tunnels

Daniel J. Bernstein, Tung Chou, Kai-Min Chung, Tanja Lange, Jonathan Levin, Lorenz Panny, Jon A. Solworth, Bo-Yin Yang.

Different deployment strategy

- Do not patch PQC onto existing network protocols, but add a new layer with superior security.

Different deployment strategy

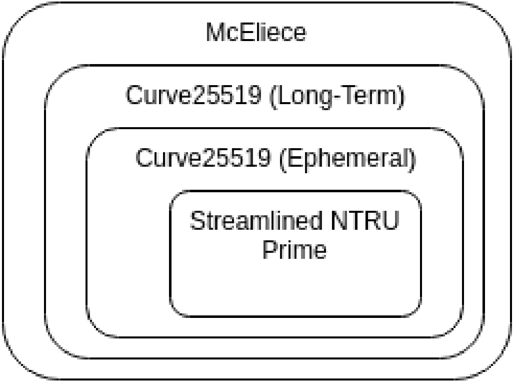
- Do not patch PQC onto existing network protocols, but add a new layer with superior security.
- Can be gradually deployed.
- Add support for VPN-like tunnels to clients and servers.

Different deployment strategy

- Do not patch PQC onto existing network protocols, but add a new layer with superior security.
- Can be gradually deployed.
- Add support for VPN-like tunnels to clients and servers.
- PQConnect is designed for security, handshake and ratcheting proven using Tamarin prover (formal verification tool).
- Use Curve25519 (pre-quantum) and Classic McEliece (conservative PQC) for long-term identity keys.
- Use Curve25519 (pre-quantum) and Streamlined NTRU Prime (PQC) for ephemeral keys.

PQConnect handshake: Nesting schemes

Most conservative system on the outside.



Attacker can see long-term Curve25519 identity key,
can break it with a quantum computer,
but cannot obtain DH value as client's share is wrapped.

1-RTT handshake

Initiator		Responder
$\mathtt{spk}_R^{\mathtt{McEliece}}, \mathtt{spk}_R^{\mathtt{x25519}}$		$\mathtt{ssk}_R^{\mathtt{McEliece}}, \mathtt{ssk}_R^{\mathtt{x25519}}$

$$\mathtt{epk}_I^{\mathtt{x25519}}, \mathtt{esk}_I^{\mathtt{x25519}} \leftarrow \text{x25519.keygen}()$$
$$\mathtt{epk}_I^{\mathtt{SNTRUP}}, \mathtt{esk}_I^{\mathtt{SNTRUP}} \leftarrow \text{SNTRUP.keygen}()$$
$$\mathtt{msg.type} \leftarrow 0x2$$
$$\mathtt{c}_0, \mathtt{k}_0 \leftarrow \text{McEliece.Encap}(\mathtt{spk}_R^{\mathtt{McEliece}})$$
$$\mathtt{C}_I \leftarrow \mathtt{k}_0, \mathtt{k}_0 \leftarrow \epsilon, \mathtt{H}_I \leftarrow \mathtt{c}_0$$
$$\mathtt{c}_1^* \leftarrow \text{AEAD.Enc}(\mathtt{C}_I, 0, \mathtt{epk}_I^{\mathtt{x25519}}, \mathtt{H}_I)$$
$$\mathtt{k}_1 \leftarrow \text{DH}(\mathtt{esk}_I^{\mathtt{x25519}}, \mathtt{spk}_R^{\mathtt{x25519}})$$
$$\mathtt{C}_I \leftarrow \text{KDF}_1(\mathtt{C}_I, \mathtt{k}_1), \mathtt{k}_1 \leftarrow \epsilon, \mathtt{H}_I \leftarrow \text{BLAKE3}(\mathtt{H}_I || \mathtt{c}_1^*)$$
$$\mathtt{c}_2^* \leftarrow \text{AEAD.Enc}(\mathtt{C}_I, 0, \mathtt{epk}_I^{\mathtt{SNTRUP}}, \mathtt{H}_I)$$
$$\mathtt{H}_I \leftarrow \text{BLAKE3}(\mathtt{H}_I || \mathtt{c}_2^*)$$
$$\mathtt{init} \leftarrow \mathtt{msg.type} || \mathtt{c}_0 || \mathtt{c}_1^* || \mathtt{c}_2^*$$

$$\xrightarrow{\mathtt{init}}$$

$$\mathtt{k}_0 \leftarrow \text{McEliece.Decap}(\mathtt{ssk}_R^{\mathtt{McEliece}}, \mathtt{c}_0)$$
$$\mathtt{C}_R \leftarrow \mathtt{k}_0, \mathtt{k}_0 \leftarrow \epsilon, \mathtt{H}_R \leftarrow \mathtt{c}_0$$
$$\mathtt{epk}_I^{\mathtt{x25519}} \leftarrow \text{AEAD.Dec}(\mathtt{C}_R, 0, \mathtt{c}_1^*, \mathtt{H}_R)$$
$$\mathtt{k}_1 \leftarrow \text{DH}(\mathtt{ssk}_R^{\mathtt{x25519}}, \mathtt{epk}_I^{\mathtt{x25519}})$$
$$\mathtt{C}_R \leftarrow \text{KDF}_1(\mathtt{C}_R, \mathtt{k}_1), \mathtt{k}_1 \leftarrow \epsilon, \mathtt{H}_R \leftarrow \text{BLAKE3}(\mathtt{H}_R || \mathtt{c}_1^*)$$
$$\mathtt{epk}_I^{\mathtt{SNTRUP}} \leftarrow \text{AEAD.Dec}(\mathtt{C}_R, 0, \mathtt{c}_2^*, \mathtt{H}_R)$$
$$\mathtt{H}_R \leftarrow \text{BLAKE3}(\mathtt{H}_R || \mathtt{c}_2^*)$$
$$\mathtt{msg.type} \leftarrow 0x3$$
$$\mathtt{epk}_R^{\mathtt{x25519}}, \mathtt{esk}_R^{\mathtt{x25519}} \leftarrow \text{x25519.keygen}()$$
$$\mathtt{c}_3^* \leftarrow \text{AEAD.Enc}(\mathtt{C}_R, 1, \mathtt{epk}_R^{\mathtt{x25519}}, \mathtt{H}_R)$$
$$\mathtt{k}_2 \leftarrow \text{DH}(\mathtt{esk}_R^{\mathtt{x25519}}, \mathtt{epk}_I^{\mathtt{x25519}})$$
$$\mathtt{C}_R \leftarrow \text{KDF}_1(\mathtt{C}_R, \mathtt{k}_2), \mathtt{k}_2 \leftarrow \epsilon, \mathtt{H}_R \leftarrow \text{BLAKE3}(\mathtt{H}_R || \mathtt{c}_3^*)$$
$$\mathtt{c}_4, \mathtt{k}_3 \leftarrow \text{SNTRUP.Encap}(\mathtt{epk}_I^{\mathtt{SNTRUP}})$$
$$\mathtt{c}_5^* \leftarrow \text{AEAD.Enc}(\mathtt{C}_R, 0, \mathtt{c}_4, \mathtt{H}_R)$$
$$\mathtt{C}_R \leftarrow \text{KDF}_1(\mathtt{C}_R, \mathtt{k}_3), \mathtt{k}_3 \leftarrow \epsilon, \mathtt{H}_R \leftarrow \text{BLAKE3}(\mathtt{H}_R || \mathtt{c}_5^*)$$
$$\mathtt{tunnelID}, \mathtt{T}_I, \mathtt{T}_R \leftarrow \text{KDF}_3(\mathtt{C}_R, \mathtt{H}_R)$$
$$\mathtt{resp} \leftarrow \mathtt{msg.type} || \mathtt{c}_3^* || \mathtt{c}_5^*$$

$$\xleftarrow{\mathtt{resp}}$$

$$\mathtt{epk}_R^{\mathtt{x25519}} \leftarrow \text{AEAD.Dec}(\mathtt{C}_I, 1, \mathtt{c}_3^*, \mathtt{H}_I)$$
$$\mathtt{k}_2 \leftarrow \text{DH}(\mathtt{esk}_I^{\mathtt{x25519}}, \mathtt{epk}_R^{\mathtt{x25519}})$$
$$\mathtt{C}_I \leftarrow \text{KDF}_1(\mathtt{C}_I, \mathtt{k}_2), \mathtt{k}_2 \leftarrow \epsilon, \mathtt{H}_I \leftarrow \text{BLAKE3}(\mathtt{H}_I || \mathtt{c}_3^*)$$
$$\mathtt{c}_4 \leftarrow \text{AEAD.Dec}(\mathtt{C}_I, 0, \mathtt{c}_5^*, \mathtt{H}_I)$$
$$\mathtt{k}_3 \leftarrow \text{SNTRUP.Decap}(\mathtt{esk}_I^{\mathtt{SNTRUP}}, c_4)$$
$$\mathtt{C}_I \leftarrow \text{KDF}_1(\mathtt{C}_I, \mathtt{k}_3), \mathtt{k}_3 \leftarrow \epsilon, \mathtt{H}_I \leftarrow \text{BLAKE3}(\mathtt{H}_I || \mathtt{c}_5^*)$$
$$\mathtt{tunnelID}, \mathtt{T}_I, \mathtt{T}_R \leftarrow \text{KDF}_3(\mathtt{C}_I, \mathtt{H}_I)$$
$$\mathtt{epk}_R^{\mathtt{x25519}}, \mathtt{epk}_I^{\mathtt{x25519}}, \mathtt{esk}_I^{\mathtt{x25519}}, \mathtt{epk}_I^{\mathtt{SNTRUP}}, \mathtt{esk}_I^{\mathtt{SNTRUP}}, \mathtt{c}^4, \mathtt{C}_I, \mathtt{H}_I \leftarrow \epsilon$$

$$\mathtt{epk}_I^{\mathtt{25519}}, \mathtt{epk}_I^{\mathtt{SNTRUP}}, \mathtt{epk}_R^{\mathtt{x25519}}, \mathtt{esk}_R^{\mathtt{x25519}}, \mathtt{C}_R, \mathtt{H}_R \leftarrow \epsilon$$

Key ratchet advances by message and time

c_0 is the initial key.
Immediately advance ratchet in 3 ways:

- New epoch master key: c_1.
- New branch keys: $c_{0,1}, c_{0,2}$.
- New message key: $c'_{0,1}$.

Delete key as soon as no longer needed.
Message keys can deal with delayed transmissions.