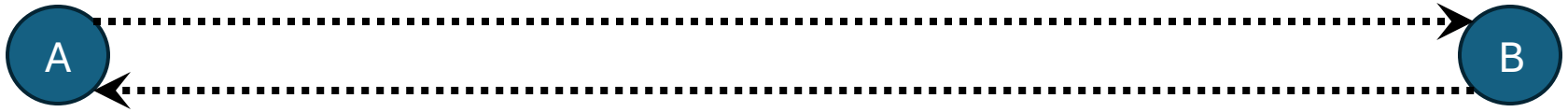
Space Security and QUIC

Benjamin Dowling	Kings College London
Britta Hale	NPS
Konrad Kohbrok	Phoenix R&D
Raphael Robert	Phoenix R&D
Xisen Tian	NPS
Bhagya Wimalasiri	Sheffield University

Space & Interplanetary Communication

Terrestrial Communication	Interplanetary Communication
Continuous availability of links and devices	Discontinuous availability
Low latency and fast RTT, e.g., RTT between two points is approximately 100 to 300 ms	Latency
Low packet loss rate	Packet loss, attenuation, jamming potential
Bi-directional data links	Data links can be simplex
Ubiquitous client-server model	Handshakes are unreliable/costly

Session establishment handshakes can be unreliable: packet loss incurs further delays in key establishment



QUIC

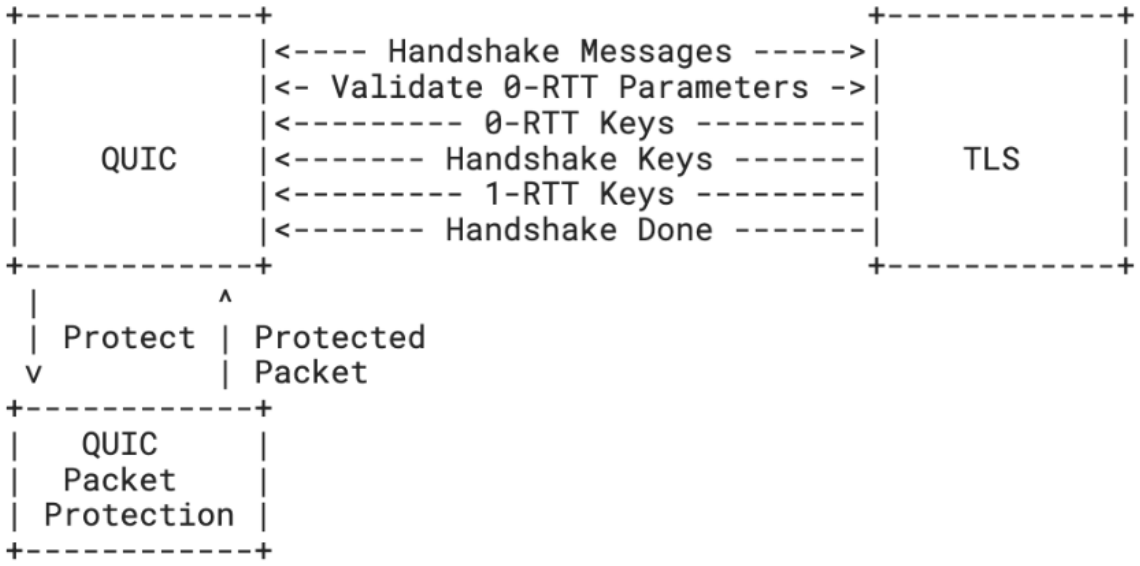
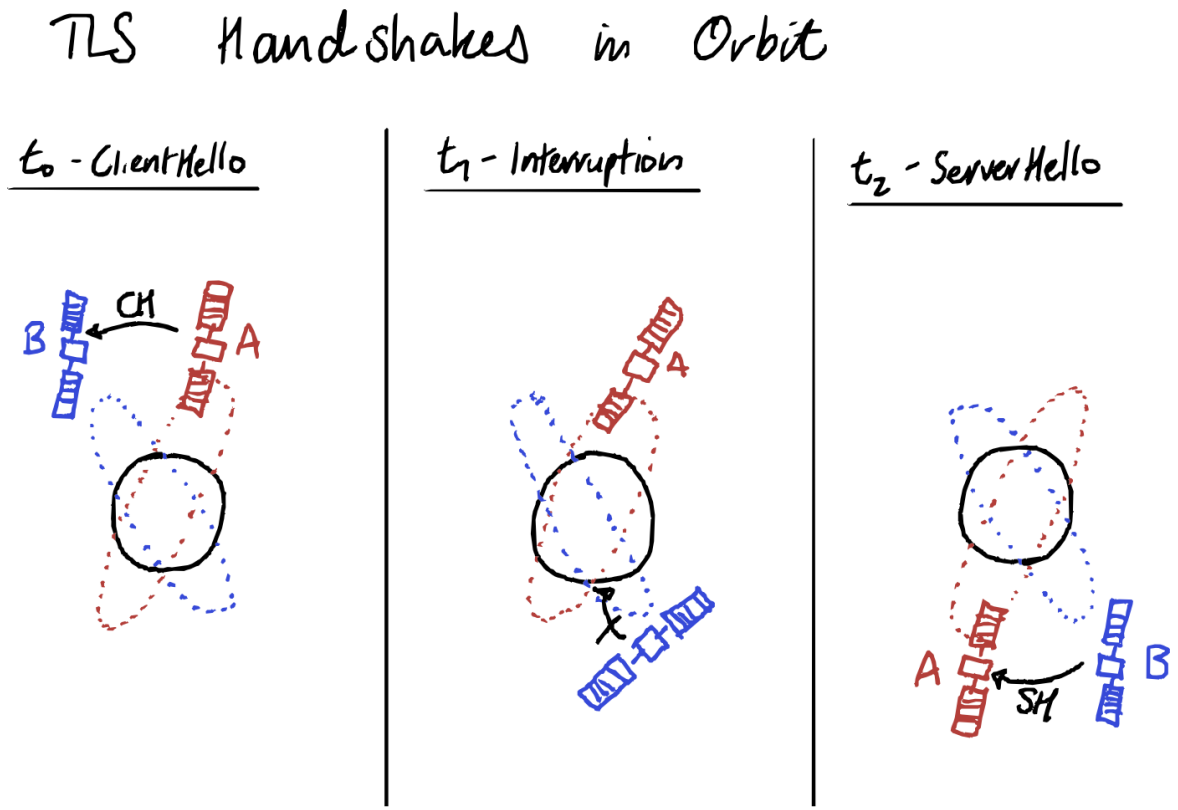


Figure: Using TLS to Secure QUIC (RFC9001)

QUIC: *Interactive integrated key establishment, secure channel and transport protocol*

- Uses TLS 1.3 handshake and authenticated encryption (requires 1-RTT)
- 0-RTT transmission of encrypted data
- Forward secrecy with deterministic key updates: $\texttt{secret}_{n+1} = \text{HKDF-Expand}(\texttt{secret}_n, \texttt{quic ku}, \emptyset, \text{H\_len})$
- Ubiquitous use in terrestrial internet apps

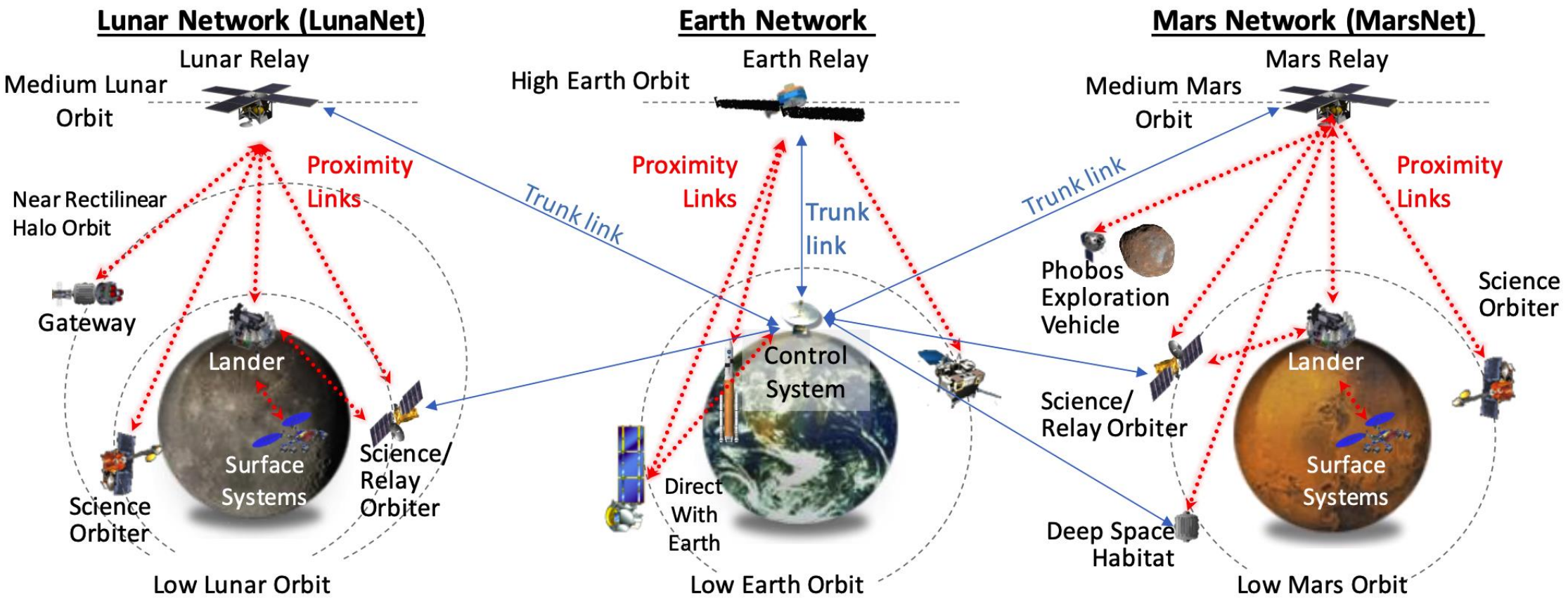
Possible Scenarios for QUIC in Space



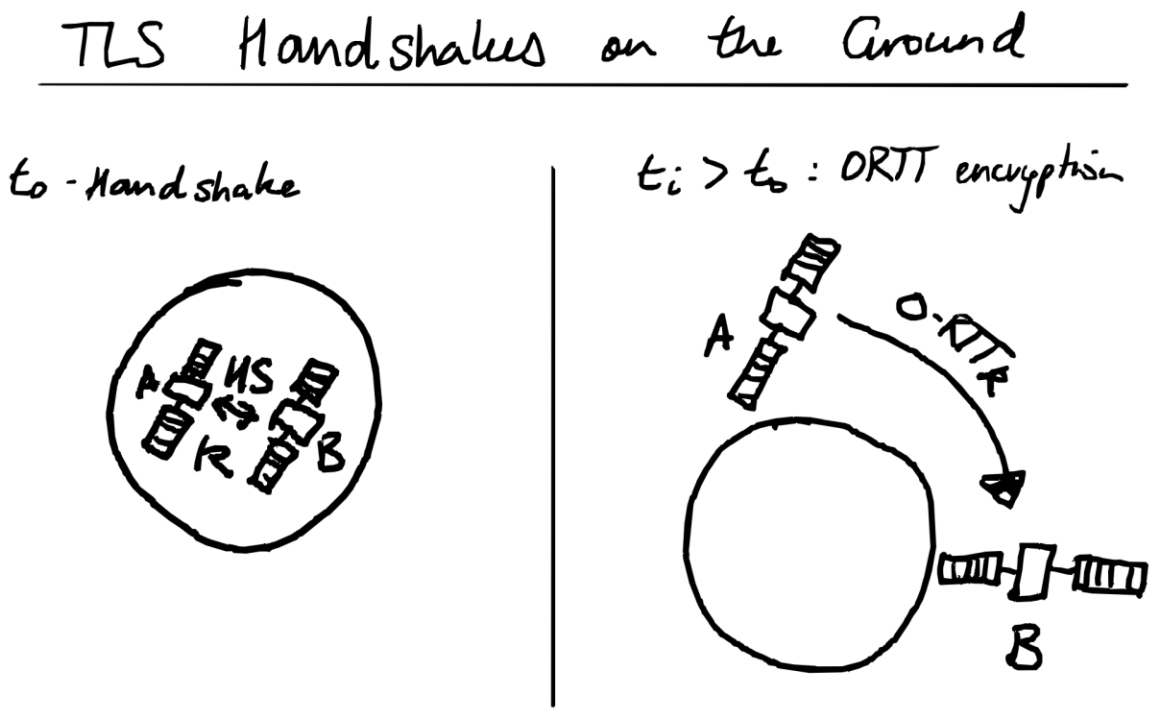
Issues: Use of synchronous handshakes in orbit is inherently challenging with link disruptions preventing useful data to be sent during setup. But the use of asynchronous key agreements allows for data to be sent immediately.

LunaNet and Planetary Networks

Planetary Networks: Earth, Moon and Mars - One Architecture

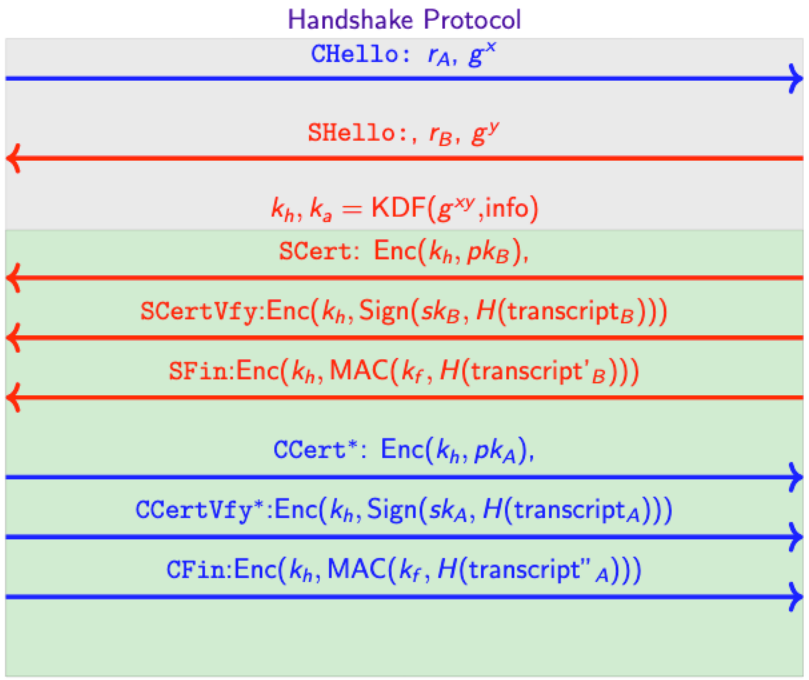


Possible Scenarios for QUIC in Space



Issues: Use of only 0-RTT encryption has known issues (see RFC 8446): key exhaustion and loss of Forward Secrecy if PSK is revealed

Issues:
QUIC's TLS handshake was designed for synchronous communication in stable environments where full RTT is the norm



***TLS was not designed for the asynchronous setting

- QUIC's TLS handshake **requires synchronous communication and 1RTT**
- **Interactive contribution *before* a secure channel**: Alice and Bob must both contribute to establishment of session key before encryption
- **Incentivizes bad security practice in space** e.g.,
  - long lived session keys
  - 0-RTT
- **High Risk in Low Latency**: 0-RTT mode has known issues (see RFC 8446)
- Poor performance under repeated **connection loss**
- **Is about to get a lot more bandwidth and packet loss risk** to 1RTT under PQC
- **No Post-Compromise Security:** Future Confidentiality lost once keys are compromised
- **Key Update**: scales poorly and does not refresh randomness

Is QUIC the right choice here?
→ Security under the use case aside, it has nice transport functionalities

Is TLS the right choice here?
→ QUIC handshake currently “slots in” the TLS handshake, which is where security functionality does not align to space requirements

How do we do security “right” for the challenges of space?

Continuous Key agreement (CKAs), a.k.a. Ratcheted Key Exchange

Functionality

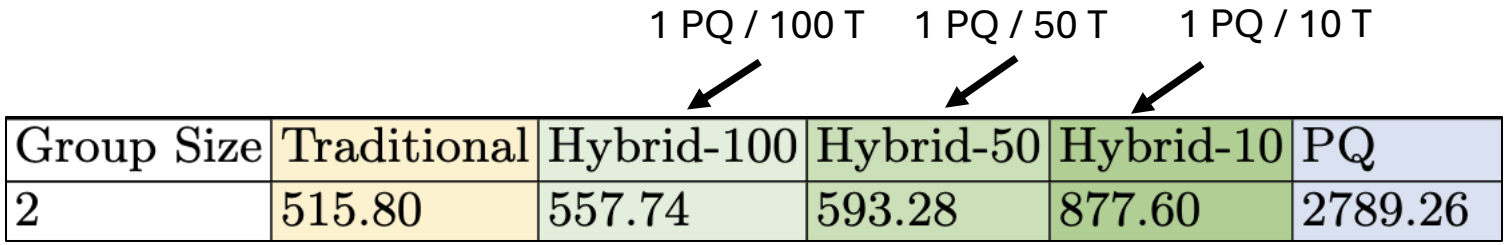
- Instead of requiring per-session handshakes, establish a session once and then **ratchet forward on demand/scheduled**
  - Not just a symmetric key ratchet (Forward Secrecy Only)
  - Also introduces fresh randomness (provides Post Compromise Security)
  - By configuration key ratchet can control for resumption, making **0-RTT resumption consecutive connections that offers contributive randomness**
- Asynchronous, key updates can be **unidirectional**
- Sessions are long-lasting, and can evolve updates over even years
- Low bandwidth & complexity
- Could also "slot in" like the TLS handshake does
- (Optional) Can handle groups of devices and dynamic group membership

Security

- E2E Authenticity and Confidentiality Adversarial Model
- Corruption: Can heal from state compromise
- Adaptive and Active Adversary
- Forward and Post-Compromise Security
- Supports **post-quantum efficiently**

Example PQ overhead of MLS/RFC 9420:

Average **bytes** / epoch for control messages (key update and associated messages) across 500 epochs



Group Size	Traditional	Hybrid-100	Hybrid-50	Hybrid-10	PQ
2	515.80	557.74	593.28	877.60	2789.26

*Improvements in both bytes on the wire and CPU cycles

This is vs. TLS at 17-19Kb for key update. *

*Research has shown that QUIC, based on TLS, outperforms TLS by ~52% with RSA but by as little as ~2.5% with NIST algorithms. https://ieeexplore.ieee.org/document/10188358

Tests by Noah Greene

Does this mean the need for a new standardized protocol for CKAs?

- No! One has already been standardized (RFC 9420, a.k.a. MLS)

Does this require security at a different network layer?

- No! RFC 9420 offers a key exchange. It does include an encryption option for using those keys at the application layer, but this is not intrinsic to the protocol.
- Just like how the TLS handshake is ‘slotted into’ QUIC, we can take that out and slot-in the MLS/RFC 9420 handshake, thus preserving all the other functionalities of QUIC while ensuring security suitable to the space use cases.

Proposal and more information:

Action to add the MLS/RFC 9420 handshake as an extension option for QUIC.

Use in space.

- Draft: https://www.ietf.org/archive/id/draft-tian-quic-quicmls-00.html
- MLS RFC 9420: https://www.rfc-editor.org/rfc/rfc9420.html
- MLS Architecture RFC 9750: https://www.rfc-editor.org/rfc/rfc9750.html

Summary

- MLS is an asynchronous protocol that is standardized, analyzed, and available for use today
- QUIC(TLS) is synchronous and is non-optimal for space
- Key Recovery from disruptions requires additional handshakes with QUIC. It requires no handshake interaction in MLS.
- Security
  - Post Compromise Security is designed into MLS but not QUIC
  - Post Quantum performance is optimized with MLS but not QUIC
    - PQ Authentication is efficient and available with MLS now

Contact

- Benjamin Dowling benjamin.dowling@kcl.ac.uk
- Britta Hale britta.hale@nps.edu
- Konrad Kohbrok konrad.kohbrok@datashrine.de
- Raphael Robert ietf@raphaelrobert.com
- Xisen Tian xisen.tian1@nps.edu
- Bhagya Wimalasiri b.m.wimalasiri@sheffield.ac.uk

Reference Slides

(Deep) space would be optimized with handshake functionality support for…

- **Asynchronous communication, esp. at further distances**
- **Secure channels that can update/evolve without latency**
- **Good security guarantees even in low latency**
- **Handle delay**
- **Has analysis assumptions that match to the use**
- Ability to support PQC

QUIC is confined by….

- QUIC's TLS handshake **requires synchronous communication and 1RTT**
- **Interactive contribution *before* a secure channel**: Alice and Bob must both contribute to establishment of session key before encryption
- **High Risk in Low Latency**: 0-RTT mode has known issues (see RFC 8446)
- **Incentivizes bad security practice in space** e.g.,
  - long lived session keys
  - 0-RTT
- Is about to get a lot more bandwidth risk and packet loss risk to 1RTT under PQC
- **Key Update**: scales poorly and does not refresh randomness