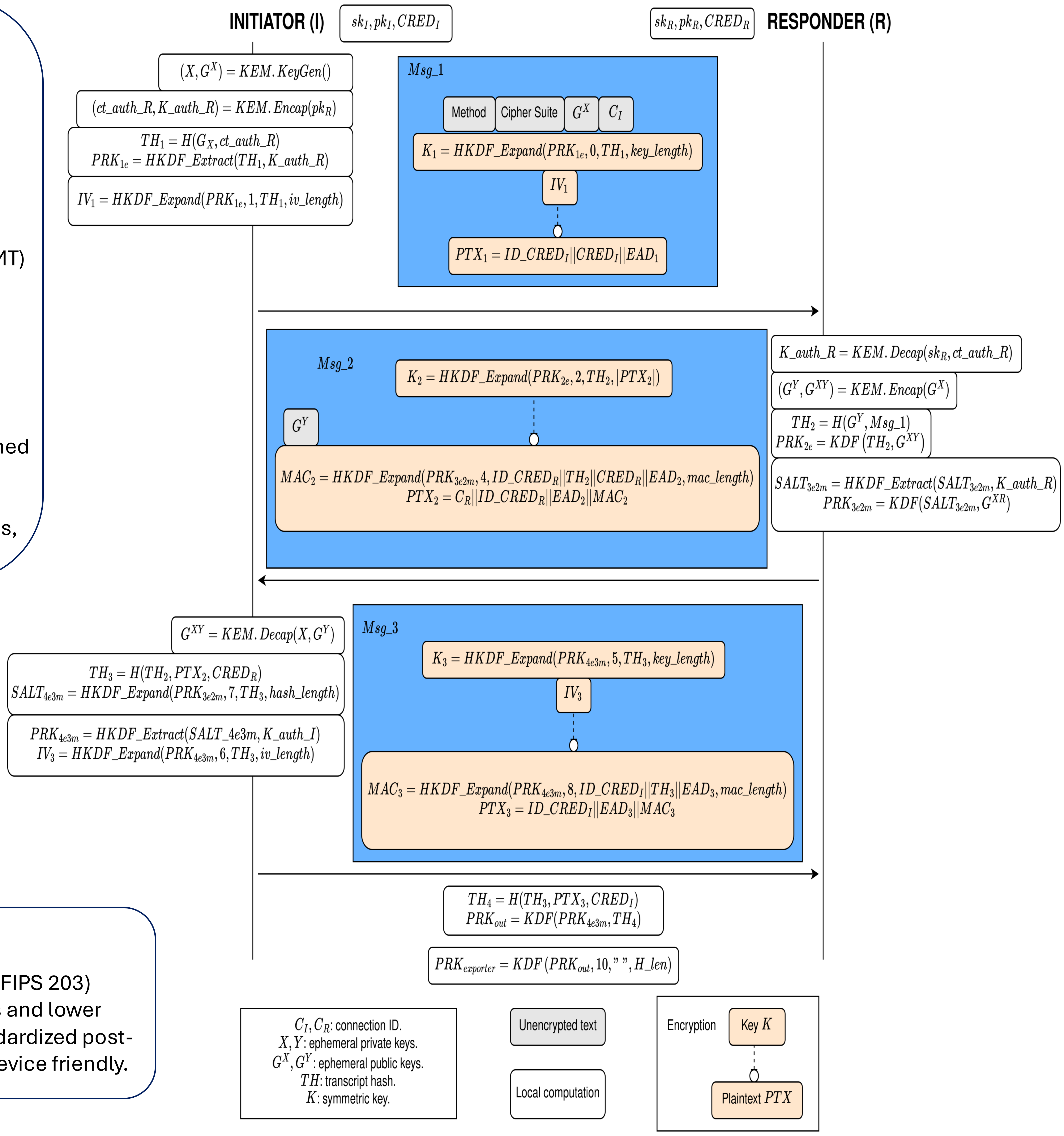
KEMEDHOC*: A formally verified implementation of quantum-resistant EDHOC key exchange protocol

(Work in progress)

Cuong Nguyen, Sampo Sovio

Background:

- Ephemeral Diffie-Hellman Over COSE (**EDHOC**) [RFC 9528] is a lightweight authenticated key exchange.
- **EDHOC** demonstrates low overhead, especially in message size, in resource-constrained settings compared to other lightweight competitors, such as DTLS.
- **Z3** is a Satisfiability Modulo Theories (SMT) solver that automatically check logical formulas for satisfiability under various theories. Z3 is used to discharge proof obligations encoded by F*,
- **F*** is a proof-oriented language with rich type annotations that encode logical properties. **Low*** is a subset of F* designed for low-level code.
- **ProVerif** is an automated tool, based on Dolev-Yao model and applied pi-calculus, for protocol formal verification.

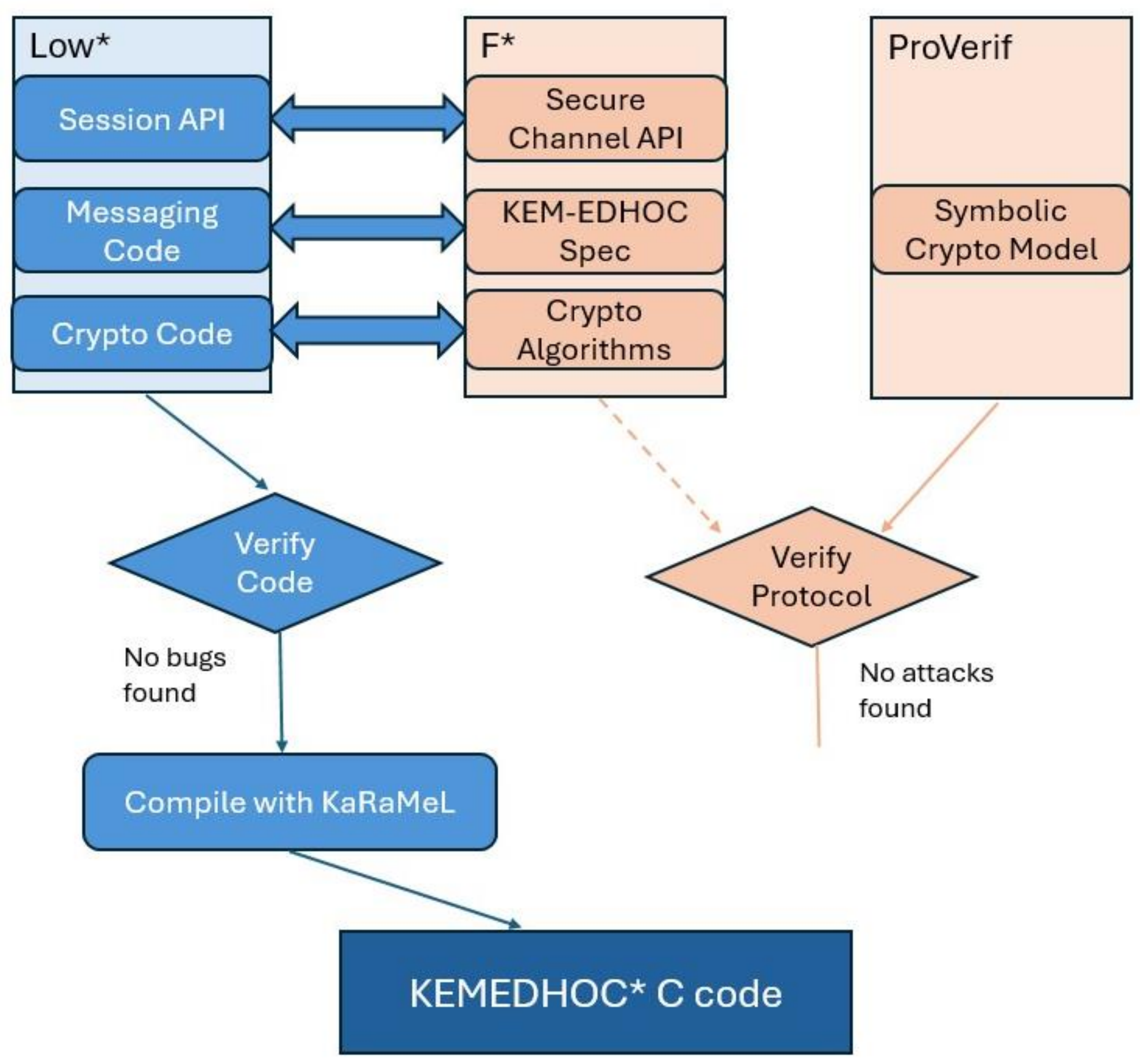


Problems:

- The standardized EDHOC protocol [RFC 9528] is **not quantum-safe**.
- Reference implementations of EDHOC are not formally verified (existing security analysis works only verify at the design level) -> **lack of end-to-end verification.**

Why ML-KEM is selected?

- ML-KEM is officially standardized by NIST (FIPS 203)
- ML-KEM introduces reasonable overheads and lower memory footprint compared to other standardized post-quantum schemes -> more constrained-device friendly.



Security Goals

Design level:

- Confidentiality
- Mutual key authentication
- Session key uniqueness
- Identity protection
- Downgrade protection
- *Quantum-proof*

Implementation level:

- Functional correctness
- Memory safety
- Side-channel resistance
- API-misuse resilience

Future work:

- Double-test with static analysis tools.
- Do benchmarking and compare to unverified implementations, e.g. uEDHOC.

Approach

Design level:

- Design a new KEM-based, signature-free authentication method which uses static and ephemeral KEM keys for authentication and shared key derivation respectively.
- Formally verify the symbolic model of new design using ProVerif.

Implementation level:

- Write and formally verify the high-level computational model (F*) of KEMEDHOC.
- Write the machine-level model (Low*) that correctly links to the F* model, guarantees memory safety and resistance to classes of side-channel attacks.
- Extract to C code using KaRaMel.

Helsinki System Security Lab (HSSL)

HSSL drives renewal and mastery in the field of platform and device related security technologies, especially for Huawei consumer devices such as mobile phones, laptops, televisions and automotive. We do research in topics such as hardware-assisted isolation and integrity, as well as in operating system protection (hypervisor, TEE, secure enclaves and kernel hardening). We also carry expertise in cryptography and systems security functionality such as device key management (PKI), device attestation and key-store solutions.

HUAWEI