INTER IIT
TECH MEET 14.0

LOW PREP

Qtrino

PQC-DTLS for Bare-Metal RISC-V IoT Devices

INTER IIT TECH MEET 14.0

Introduction

As the world moves toward the adoption of post-quantum cryptography (PQC), there is a growing need to integrate modern communication security protocols such as TLS 1.3 and DTLS 1.3 with PQC algorithms to ensure secure communication in the quantum era. These protocols play a critical role in establishing confidentiality, integrity, and authentication for data in transit across enterprise, industrial control systems (ICS) and operational technology (OT) in the post quantum era. Implementing PQC enabled TLS and DTLS 1.3 is therefore an important step toward future-proofing secure communication across networks, devices, and critical infrastructure.

About Us

QTrino Labs Private Limited is a deep tech cybersecurity company dedicated to developing cost-effective, cutting edge, quantum-safe solutions that safeguard businesses and governments from emerging quantum threats. Founded in August 2023, QTrino Labs specializes in simplifying complex post quantum cryptography, enabling organizations of all sizes to seamlessly transition to quantum resilient security. We are officially recognized under Startup India, Startup Bihar, and the Department of Telecommunications (TEC), Government of India, and are active members of the National Quantum Mission and the Global PKI Consortium positioning us at the forefront of India's and the world's transition to post quantum cryptography.

Motivation

Implementing TLS 1.3 and DTLS 1.3 with post-quantum cryptographic (PQC) algorithms in Industrial Control Systems (ICS), Operational Technology (OT), and IoT embedded devices poses a significant engineering challenge due to severe hardware and computational constraints. These systems typically operate on low-power processors with limited memory, bandwidth, and energy availability, often without a full operating system or hardware acceleration for cryptographic operations.

INTER IIT TECH MEET 14.0

While PQC algorithms provide stronger security against quantum threats, they introduce larger key sizes, increased computational complexity, and higher communication overhead compared to classical algorithms such as RSA or ECC. Integrating these algorithms into TLS 1.3 and DTLS 1.3 handshakes in such constrained environments demands meticulous optimization of memory usage, execution cycles, and protocol handling. Achieving quantum-safe, standards-compliant communication at the bare-metal or BIOS level without compromising performance or real-time responsiveness remains one of the most pressing engineering challenges in securing embedded systems for the post-quantum era.

Problem Statement

Description

The problem is to design and demonstrate a post quantum secure DTLS 1.3 communication channel with full mutual authentication between a server and a RISC-V–based IoT device operating on a bare-metal environment (BIOS-level, without an operating system). The task involves integrating suitable post-quantum cryptographic (PQC) algorithms into the DTLS 1.3 protocol, optimizing cryptographic operations for constrained RISC-V hardware, and ensuring compatibility with standard DTLS 1.3 handshake procedures. The implementation must establish a secure session while maintaining efficiency in terms of latency, memory usage, and computational overhead, addressing the unique challenges of deploying PQC based secure communication in low power, resource limited embedded systems.

INTER IIT TECH MEET 14.0

REQUIREMENTS

Participants will need the following:

- Verilator installed on their PC
- RISC-V GCC toolchain (for compiling bare-metal firmware)
- LiteX software environment
- wolfSSL + wolfCrypt library (source code)
- Python 3 and standard build tools
- Makefile (provided)
- Linker script (provided)
- wolfSSL/wolfCrypt configuration file (provided)
- Git (for cloning repositories)
- Basic Linux environment (Ubuntu recommended)
- Wireshark for observing the DTLS 1.3 handshake
- Ethernet or UART loopback setup for testing communication
- QEMU/Spike (if they want to run functional RISC-V tests)

NOTE FOR PARTICIPANTS

Refer to **this README** which explains the following details :

- how to install each required tool and library,
- how to set up the LiteX + Verilator simulation environment,
- how to compile and run RISC-V bare-metal code, and
- how to integrate wolfSSL/wolfCrypt for DTLS 1.3 testing.

This README will serve as your step-by-step guide throughout the challenge.

INTER IIT TECH MEET 14.0

DELIVERABLES

Each participant/team must submit the following:

1. **RISC-V Bare-Metal Firmware**
   a. Working firmware running on the LiteX + Verilator simulated RISC-V SoC
   b. Firmware must initiate a **PQC-DTLS 1.3 client handshake** with the server
   c. wolfSSL/wolfCrypt integrated and configured for bare-metal operation
   d. Clean, readable and well commented/documented C source code.

2. **PQC-DTLS 1.3 Communication Demo**
   a. Demonstration of successful one-to-one DTLS 1.3 communication (client-server)
   b. Use of at least one PQC KEM (e.g., Kyber) for key exchange
   c. Use of a PQC signature scheme (e.g., Dilithium) for authentication (if required)
   d. Packet capture (e.g., Wireshark screenshot) showing DTLS 1.3 handshake completion

3. **Build + Run Instructions**
   a. A short README explaining:
      i. how to compile the RISC-V firmware
      ii. how to run it in the LiteX + Verilator simulation environment
      iii. how to start the server and reproduce the DTLS handshake
   b. Commands/scripts required to build and test the project

4. **Configuration Files**
   a. wolfSSL/wolfCrypt config header used
   b. Linker script
   c. Makefile (modified as required by participant)
   d. Any additional lightweight modules written by the participant

INTER IIT TECH MEET 14.0

5. **Short Technical Report (2–3 pages)**
    a. Problem understanding
    b. Architecture and design approach
    c. PQC algorithm choices and rationale
    d. Challenges faced & how they were solved
    e. Security considerations
    f. Performance measurements (latency, memory footprint, ROM/RAM usage)
    g. Support for session resumption
    h. Additional optimisations for low-power RISC-V hardware
    i. Integrating custom entropy sources (TRNG/PRNG)

EVALUATION CRITERIA

1. **Latency**
    a. DTLS 1.3 handshake latency (client - server)
    b. Time taken for full session establishment
    c. Processing time for PQC key exchange and authentication
    d. Low oscillations and predictable corrections

2. **Throughput**
    a. Data transmission throughput after session establishment
    b. Efficiency of cryptographic operations under load
    c. Stability of throughput across multiple message exchanges

3. **Memory Usage**
    a. ROM footprint (final firmware size after compiling)
    b. RAM usage during handshake + active session
    c. wolfSSL/wolfCrypt PQC configuration memory efficiency
    d. Optimization in stack/heap usage for a constrained RISC-V system

4. CPU Utilization / Compute Efficiency

a. Number of CPU cycles consumed during:

i. PQC key exchange

ii. DTLS handshake

iii. Data encryption/decryption

b. Efficient use of RISC-V instructions, minimal overhead

5. Implementation Correctness

a. Successful PQC-DTLS 1.3 session establishment

b. Correct integration of PQC KEM + signature algorithms

c. Proper DTLS record formatting, retransmission handling, alerts, etc.

6. Resource Optimization

a. Minimal code bloat

b. Removing unnecessary wolfSSL modules

c. Efficient linker script & section placement

d. Performance on bare-metal LiteX simulation

7. Reliability Under Constraints

a. Clean run without handshake failures

b. Proper error handling

c. Stable session continuity for multiple messages

INTER IIT TECH MEET 14.0

Judging Criteria

In the first round, judges will evaluate papers using a **standardised rubric** to ensure **fairness and consistency** in scoring.
Scores from each judge will be **normalised and aggregated** to determine the **top three papers** in each category.

Category	Description	Weightage
Latency	DTLS 1.3 handshake time and PQC operation delay	25%
Throughput	Sustained data transfer rate over DTLS 1.3	20%
Memory Usage	ROM/RAM footprint and optimization on RISC-V	15%
Correctness & Compliance	Valid PQC-DTLS 1.3 implementation, no errors/malformed packets	15%
Reliability & Stability	Consistent handshakes, error handling, stable operation	10%
Technical Report	Quality of explanation, architecture clarity, rationale, challenges	10%
Presentation	Clarity , Story And Delivery	5%

INTER IIT TECH MEET 14.0

RESOURCES

1. https://github.com/enjoy-digital/litex - Litex build
2. https://datatracker.ietf.org/doc/rfc9147/ - DTLS 1.3 Documentation
3. https://csrc.nist.gov/projects/post-quantum-cryptography - Post Quantum Cryptography
4. https://www.wolfssl.com/ - wolfssl/wolfcrypt documentation