QUANTUM-SAFE TACTICAL COMMUNICATION SYSTEM

Technical Architecture and Security Analysis

Submitted for iDEX Defence Innovation Challenge

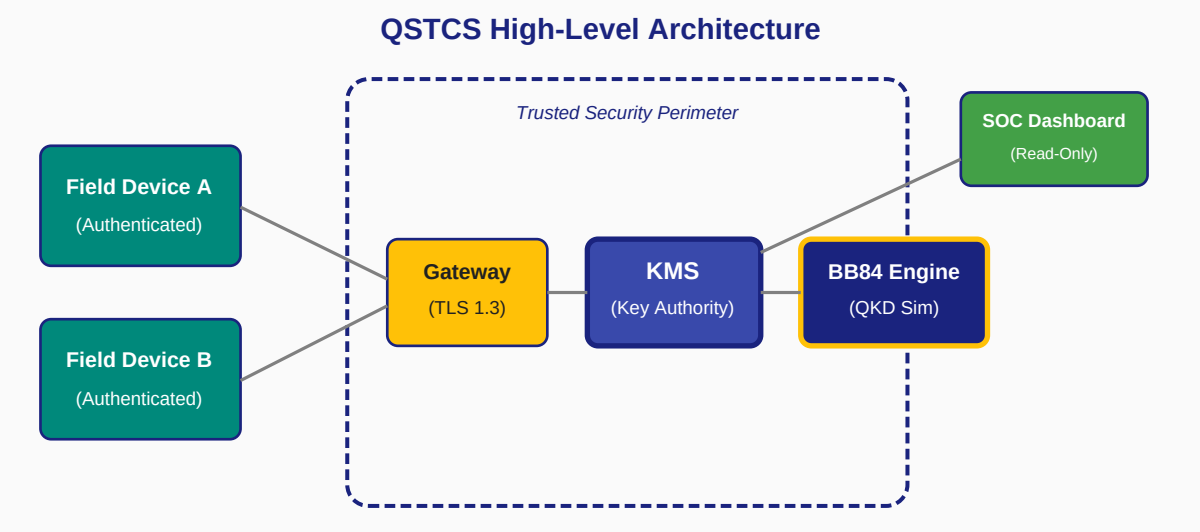

Figure 1: High-level system architecture showing trusted security perimeter and component relationships.

A software-defined quantum key distribution prototype enabling provably secure tactical communications resistant to both classical and quantum cryptanalytic attacks.

Document Version 2.0 | Classification: UNCLASSIFIED | January 2026

Contents

1. Executive Summary

The Quantum-Safe Tactical Communication System (QSTCS) is a prototype secure messaging platform designed for military field operations. Unlike conventional encryption schemes whose security relies on computational hardness assumptions vulnerable to quantum algorithms, QSTCS implements the BB84 Quantum Key Distribution (QKD) protocol, which derives its security guarantees from the fundamental laws of quantum mechanics.

The system provides three critical capabilities: (1) generation of cryptographic keys with information-theoretic security, (2) real-time detection of eavesdropping attempts through Quantum Bit Error Rate (QBER) monitoring, and (3) authenticated encryption of tactical messages using AES-256-GCM with quantum-derived keys. This design addresses the "harvest now, decrypt later" threat posed by adversaries stockpiling encrypted traffic for future quantum decryption.

> *Key Innovation: Software-defined QKD simulation enabling rapid prototyping and seamless migration to hardware QKD infrastructure when deployed.*

2. Threat Landscape and Motivation

2.1 The Quantum Computing Threat

Current asymmetric cryptographic systems (RSA, ECDH, DSA) rely on the computational intractability of integer factorization and discrete logarithm problems. Shor's algorithm, executable on a sufficiently powerful quantum computer, solves these problems in polynomial time, rendering RSA-2048 and ECDH-256 effectively broken. While fault-tolerant quantum computers capable of running Shor's algorithm at scale do not yet exist, intelligence agencies assess their emergence within 10-15 years.

2.2 Harvest Now, Decrypt Later (HNDL)

Adversaries are actively intercepting and storing encrypted communications with the intent to decrypt them once quantum capabilities mature. For classified military communications with long-term sensitivity (strategic plans, intelligence sources, treaty negotiations), this represents an immediate operational risk. Data encrypted today using RSA or ECDH should be considered compromised against a patient adversary.

2.3 Why Quantum Key Distribution?

QKD protocols like BB84 provide information-theoretic security: their security does not depend on computational assumptions but on physical laws. Specifically, the no-cloning theorem guarantees that an eavesdropper cannot copy quantum states without detection, and measurement disturbance ensures any interception attempt introduces detectable errors. This makes QKD-derived keys provably secure against all computational attacks, including those from future quantum computers.

Security Property	RSA/ECDH	PQC (Kyber)	QKD (BB84)
Security Basis	Math Hardness	Lattice Problems	Physics Laws
Quantum Resistant	No	Believed Yes	Proven Yes
Eavesdrop Detection	None	None	Built-in (QBER)
Key Compromise	Silent	Silent	Detected
Maturity	Deployed	Standardizing	Prototype

Table 1: Comparative security properties of key establishment mechanisms.

3. System Architecture

QSTCS employs a modular architecture separating cryptographic key generation, key management, and message encryption into distinct components. This design enables independent security auditing and facilitates future hardware integration.

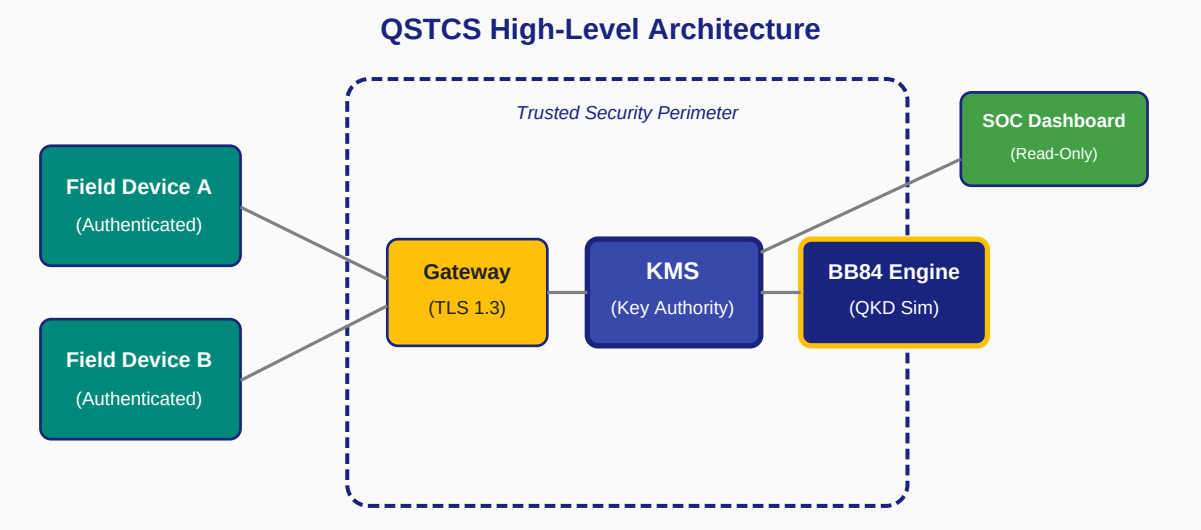

Figure 2: Component architecture with security boundary delineation.

3.1 BB84 Quantum Engine

The core cryptographic module implementing the BB84 QKD protocol. In the current prototype, quantum operations are simulated using classical randomness with physics-accurate error modeling. The engine executes the complete BB84 workflow: random bit and basis generation, qubit state preparation, basis-dependent measurement simulation, sifting, and QBER calculation. The simulation accurately models eavesdropper-induced disturbance, producing ~25% QBER under intercept-resend attacks as predicted by quantum information theory.

3.2 Key Management Service (KMS)

The trusted authority responsible for key lifecycle management. Upon receiving a key request, the KMS invokes the BB84 engine, validates the generated key against the QBER threshold (11%), and derives session keys using HKDF-SHA256. The KMS maintains session state, tracks key usage, and enforces key rotation policies. All key material is held only in volatile memory with no persistent storage.

3.3 Field Device Clients

Tactical endpoints (ruggedized laptops, mobile devices) that authenticate to the KMS and obtain session keys. Clients perform AES-256-GCM encryption/decryption locally, ensuring plaintext never leaves the device. Each message includes a unique 96-bit nonce and 128-bit authentication tag, providing both confidentiality and integrity.

3.4 Network Gateway

Message routing infrastructure connecting field devices to the KMS and to each other. The gateway handles only ciphertext and cannot access plaintext. Transport security (TLS 1.3) provides defense-in-depth, but primary security relies on the quantum-derived symmetric keys.

3.5 Security Operations Dashboard

Read-only monitoring interface displaying real-time system health: link status (secure/compromised), QBER measurements, key issuance rate, and detected attack attempts. Provides situational awareness for security

operations center (SOC) personnel without granting key access.

4. BB84 Protocol Implementation

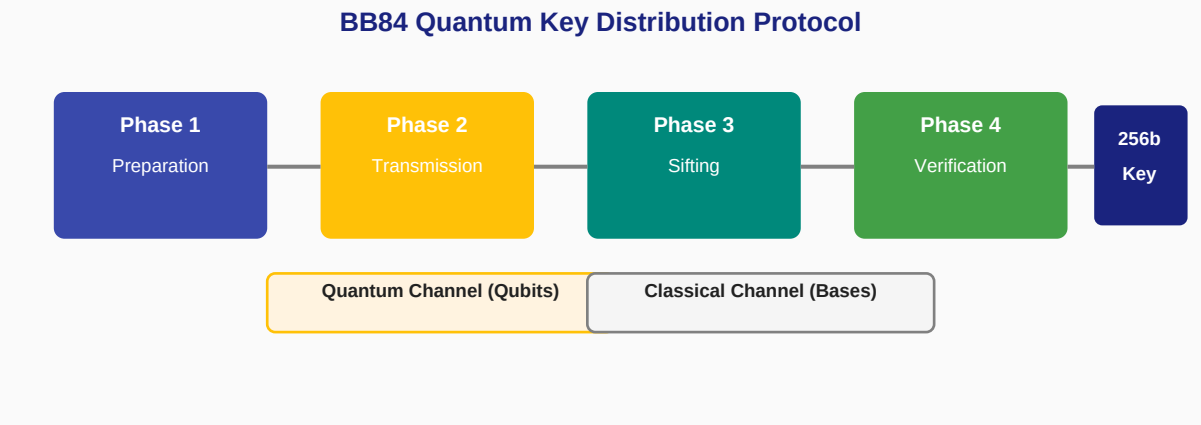

Figure 3: BB84 protocol phases from preparation through verified key output.

4.1 Protocol Phases

Phase	Alice (Sender)	Bob (Receiver)	Output
Preparation	Generate random bits b[i], bases B[i]	-	Qubit states
Transmission	Encode: |0>,|1>,|+>,|-> per b[i],B[i]	Choose random bases B'[i]	Measured bits
Sifting	Announce B[i] over classical channel	Compare B'[i], keep matches	Sifted key (~50%)
Verification	Sample subset, compute QBER	QBER < 11%: Accept	256-bit raw key

Table 2: BB84 protocol execution showing Alice and Bob operations per phase.

5. Security Analysis

5.1 Eavesdropper Detection via QBER

The security of BB84 relies on the quantum mechanical principle that measurement disturbs quantum states. When an eavesdropper (Eve) intercepts qubits, she must measure them to extract information. If Eve chooses the wrong measurement basis (50% probability), her measurement projects the qubit into a random state. When Bob subsequently measures with the correct basis, he obtains an incorrect result with 50% probability. The combined effect: Eve's interception of all qubits introduces approximately 25% error rate in the sifted key.

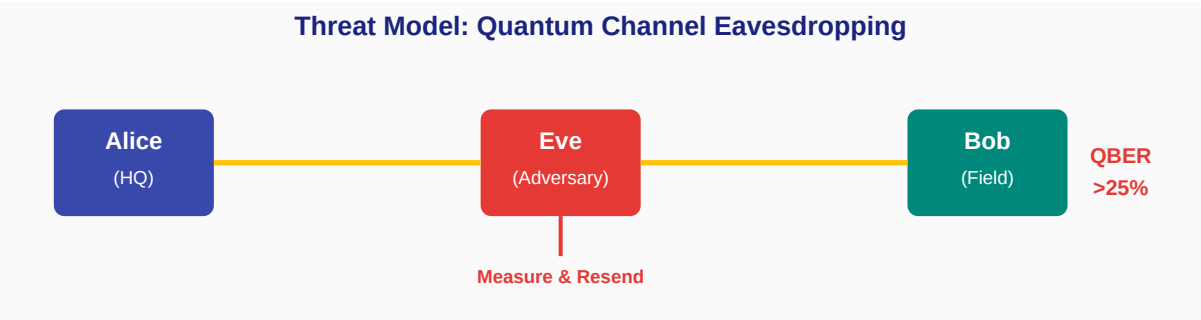

Figure 4: Intercept-resend attack model showing Eve's measurement-induced disturbance.

5.2 Security Threshold Rationale

The 11% QBER threshold is derived from information-theoretic security proofs for BB84. Below this threshold, sufficient secret key can be extracted through privacy amplification even if Eve obtained partial information. Above 11%, the protocol cannot guarantee secrecy and must abort. Our implementation conservatively refuses key issuance at QBER > 11%, alerting operators via the dashboard.

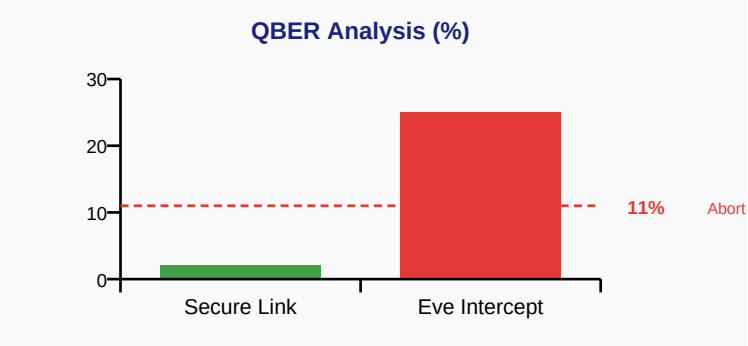

Figure 5: Measured QBER comparison between secure transmission (~2%) and active eavesdropping (~25%).

6. Operational Workflow

The following sequence illustrates a complete secure message exchange between two field units, demonstrating the integration of quantum key distribution with classical authenticated encryption.

Step	Operation	Security Property
1	Device A authenticates to KMS, requests session key	Device identity verified
2	KMS executes BB84 simulation (512 qubits)	Quantum randomness generated
3	Sifting produces ~256 correlated bits	Basis reconciliation complete
4	KMS computes QBER; verifies < 11% threshold	No eavesdropper detected
5	KMS derives AES-256 key via HKDF-SHA256	Key strengthening applied
6	Session key returned to Device A	Secure key established
7	Device A encrypts message (AES-256-GCM)	Confidentiality + integrity
8	Ciphertext transmitted via Gateway	Defense-in-depth (TLS)
9	Device B obtains session key from KMS	Symmetric key agreement
10	Device B decrypts and verifies auth tag	Message authenticity confirmed

Table 3: End-to-end message security workflow with cryptographic properties per step.

7. Technical Specifications

Component	Technology	Specification
Key Generation	BB84 Simulation	512 qubits, ~256-bit sifted key
Key Derivation	HKDF-SHA256	256-bit session key output
Symmetric Cipher	AES-256-GCM	96-bit nonce, 128-bit auth tag
QBER Threshold	Information-theoretic	11% abort threshold
Transport	TLS 1.3	Defense-in-depth only
Dashboard	Streamlit	Real-time SOC monitoring
Runtime	Python 3.8+	Cross-platform deployment

Table 4: Technical specifications and cryptographic parameters.

Source Code Structure:

```
quantum-tactical-comms/
|-- quantum_engine/bb84_simulator.py   # BB84 QKD protocol implementation
|-- kms/key_management_service.py      # Key authority and lifecycle management
|-- devices/client.py                  # Field device encryption client
|-- gateway/network_gateway.py         # Message routing infrastructure
|-- dashboard/dashboard_ui.py          # SOC monitoring interface (Streamlit)
|-- main.py                            # Console demonstration entry point
|-- tests/                             # Automated security verification tests
```

8. Conclusion and Development Roadmap

8.1 Summary of Achievements

QSTCS demonstrates a complete, functional prototype of quantum-safe tactical communications. The system successfully implements BB84 key distribution with accurate eavesdropper detection, integrates HKDF-based key derivation and AES-256-GCM encryption, and provides real-time security monitoring. Automated tests verify both normal operation (QBER ~0-3%) and attack detection (QBER ~25% triggering abort).

The software-defined architecture enables immediate deployment for training, evaluation, and operational concept development. The modular design positions the system for seamless transition to hardware QKD when tactically appropriate.

8.2 Development Roadmap

Phase	Capability	Timeline
Current	Software QKD simulation, full encryption stack	Complete
Near-term	Integration with commercial QKD hardware (QNu Labs)	6-12 months
Near-term	Hybrid PQC fallback (CRYSTALS-Kyber + BB84)	6-12 months
Mid-term	Multi-node mesh networking with key relay	12-18 months
Mid-term	Mobile platform clients (Android/iOS)	12-18 months
Long-term	Satellite QKD integration for global reach	24+ months

Table 5: Development roadmap from current prototype to operational deployment.

8.3 Strategic Value Proposition

QSTCS addresses a critical gap in defence communications: providing quantum-resistant security at the tactical edge. Unlike backbone QKD networks (e.g., QNu Labs' metropolitan deployments), QSTCS focuses on the "last mile" - bringing quantum-derived security directly to soldiers, drones, and mobile command posts. The software-defined approach enables:

1. Rapid Deployment: No specialized hardware required for initial evaluation.
2. Training and Doctrine Development: Enables personnel familiarization with quantum security concepts before hardware deployment.
3. Future-Proofing: Architecture designed for hardware QKD integration without application-layer changes.
4. Cost Efficiency: Software simulation validates operational concepts before capital investment in quantum hardware.

This document and the accompanying prototype demonstrate readiness for Phase II development and operational pilot deployment. For technical inquiries or demonstration requests, contact the development team.

--- END OF DOCUMENT ---