Quantum Signature Protocol with SKG (Alice $\rightarrow$ SKG Flow)

(Technical Document)

October 2, 2025

Abstract

This document provides a technical and concise description of the quantum signature process presented (Alice signs a state $|P\rangle$) and the sequence of messages for sending/validation with the *Secret Key Generator* (SKG). It uses QKD for key provisioning and the Quantum One-Time Pad (QOTP) for quantum confidentiality. Includes notation, operations, pseudocode, and a small example for 2 qubits.

1 Notation and Operators

- $|P\rangle = \bigotimes_{i=1}^{m} |p_i\rangle$: quantum message of m qubits prepared by Alice.
- $U(\frac{\pi}{2}, \varphi, 0)$: special U gate used in the scheme; in practice we use $U \equiv U(\frac{\pi}{2}, \varphi_A, 0)$ with φ_A secret to Alice.
- $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$: Pauli X, Z matrices.
- $QOTP_K(\cdot)$: quantum encryption via Quantum One-Time Pad using classical key $K \in \{0,1\}^{2m}$, interpreted in pairs (k_{2i-1}, k_{2i}) for qubit i. The operation per qubit is

  $$E_i = X^{k_{2i}} Z^{k_{2i-1}}.$$

  Thus,

  $$QOTP_K\big(\,|p_1\rangle \otimes \cdots \otimes |p_m\rangle\,\big) = \bigotimes_{i=1}^{m} X^{k_{2i}} Z^{k_{2i-1}} |p_i\rangle\,.$$

- T_A : symmetric key (length $\geq 2m$) established via QKD between SKG and Alice. Similarly, T_B for Bob.
- φ_A : phase parameter shared between SKG and Alice during provisioning.

- SE_A : Alice's Secure Element (stores sk_A, φ_A, performs secure operations, generates nonce and timestamp).
- $H(\cdot)$: cryptographic hash function (SHA-2/3).
- $\text{Sign}_{sk_A}(\cdot)$: classical signature performed inside SE_A (can be PQC).

2 High-level Flow Summary

1. **Provisioning (offline/prior)**: SKG and Alice perform QKD $\Rightarrow$ generate T_A. SKG generates/signs φ_A and registers $ID_A \leftrightarrow \varphi_A$.
2. **Signature (Alice)**: Alice applies $U^{\otimes m}$ on $|P\rangle$, applies $QOTP_{T_A}$ obtaining $|S\rangle$. Alice sends to Bob the triple $(|P\rangle, |S\rangle, \text{meta}, \sigma_A)$.
3. **Verification (Bob → SKG)**: Bob forwards $|S\rangle$ and metadata to SKG (encrypted with T_B). SKG decrypts $QOTP_{T_A}$ using T_A, applies $U^\dagger$ with φ_A, and recovers $|P\rangle_{rec}$; SKG returns encrypted proof to Bob.
4. Bob compares $|P\rangle_{rec}$ with his $|P\rangle$ (or with classical description of $|P\rangle$, see practical notes).

3 Formal Protocol — Enumerated Messages

Below are the main messages M0..M10.

Phase 0: Provisioning

M0.1 Physical provisioning: Alice (SE) creates ID_A, sk_A, pk_A. SKG registers ID_A and publishes certificate $Cert_A = \text{Sign}_{SKG}(ID_A, pk_A, \text{meta})$.

M0.2 QKD: SKG $\leftrightarrow$ Alice: generate T_A (classical, secret, $|T_A| \geq 2m$).

M0.3 SKG generates/shares φ_A with Alice (stored in SE).

Phase 1: Alice's Signature

M1 Alice prepares $|P\rangle$ (m qubits) and generates *nonce*, TS in SE. Forms metadata $M = (ID_A, TS, nonce, \text{meta})$.

M2 Alice applies $U^{\otimes m}(\pi/2, \varphi_A, 0)$:

$$|P'\rangle = U^{\otimes m} |P\rangle .$$

M3 Alice applies QOTP with T_A:

$$|S\rangle = QOTP_{T_A}(|P'\rangle) = \bigotimes_{i=1}^{m} X^{k_{2i}} Z^{k_{2i-1}} |p'_i\rangle .$$

M4 SE computes hash and classical signature:

$$h = H\big(\text{descr}(|S\rangle) \parallel M\big), \qquad \sigma_A = \text{Sign}_{sk_A}(h).$$

(Note: $\text{descr}(|S\rangle)$ is a label or description applicable when $|P\rangle$ is a preparable/described state.)

M5 Alice sends to Bob: $(|P\rangle, |S\rangle, M, \sigma_A, Cert_A)$ via appropriate channels.

Phase 2: Verification — Bob Queries SKG

M6 Bob validates $Cert_A$ and σ_A (using pk_A). Reject if invalid.

M7 Bob encrypts (or encapsulates) $|S\rangle, M, \sigma_A, Cert_A$ using T_B (QOTP or secure channel) and sends to SKG:

$$\text{msg}_{B\to SKG} = E_{T_B}\big(|S\rangle, M, \sigma_A, Cert_A\big).$$

M8 SKG decrypts with T_B to obtain $|S\rangle, M, \sigma_A, Cert_A$. SKG validates $Cert_A$ and σ_A.

M9 SKG decrypts $|S\rangle$ using T_A: applies $D_{T_A} = QOTP_{T_A}$ again (note $X^2 = Z^2 = I$), obtaining $|P'\rangle$, then applies $(U^\dagger)^{\otimes m}$ with φ_A to recover $|P\rangle_{rec}$.

M10 SKG responds to Bob: $E_{T_B}\big(|P\rangle_{rec}, \text{verdict}, TS_{SKG}\big)$. Bob decrypts and compares $|P\rangle_{rec}$ with $|P\rangle$ received from Alice or with classical description.

4 Important Practical Notes

- **No-cloning:** protocol assumes $|P\rangle$ is a *preparable* state by Alice and/or has a classical representation (e.g., sensor readings encoded in the computational basis). Arbitrary unknown states cannot be copied or compared without destructive measurements.
- **Handling descr($|S\rangle$):** in practical implementations, sensor $|P\rangle$ is usually classical information (CO_2 value, timestamp) encoded in computational qubits; hence description/hashing is trivial (hash of classical payload) and verification is straightforward.
- **QOTP:** requires 2 key bits per qubit; plan length and rotation of T_A accordingly.
- **Freshness/replay:** include TS and *nonce*, optionally anchor hashes in a ledger for immutable time proof.
- **SKG security:** SKG is an authority; consider threshold SKG (t-of-n) or audit/ledger to reduce central corruption risk.

- **Quantum memory:** maintaining $|S\rangle$ stable while communicating with SKG requires good fidelity; classical encoding should be used where possible.

5 Pseudocode (Algorithm)

Algorithm 1 Signature and Verification with SKG (high-level view)

procedure AliceSign($|P\rangle, SE_A, T_A, \varphi_A$)
  $M \leftarrow (ID_A, TS, nonce, meta)$
  $|P'\rangle \leftarrow U(\frac{\pi}{2}, \varphi_A, 0)^{\otimes m}\ |P\rangle$
  $|S\rangle \leftarrow QOTP_{T_A}(|P'\rangle)$
  $h \leftarrow H(\mathrm{descr}(|S\rangle) \parallel M)$
  $\sigma_A \leftarrow \mathrm{Sign}_{sk_A}(h)$ ▷ performed in SE
  return $(|P\rangle, |S\rangle, M, \sigma_A, Cert_A)$
end procedure
procedure BobVerifyViaSKG($(|P\rangle, |S\rangle, M, \sigma_A, Cert_A), T_B$)
  Validate $Cert_A$ and σ_A (using pk_A)
  if invalid **then return** REJECT
  end if
  $msg \leftarrow E_{T_B}(|S\rangle, M, \sigma_A, Cert_A)$
  send msg to SKG
  receive $resp \leftarrow E_{T_B}(|P\rangle_{rec}, verdict, TS_{SKG})$
  decrypt $resp$ with T_B
  if $|P\rangle_{rec}$ matches $|P\rangle$ **then return** ACCEPT
  elsereturn REJECT
  end if
end procedure

6 Numerical Example (2 qubits)

Suppose $m = 2$, key $T_A = (k_1, k_2, k_3, k_4)$.

- For qubit 1 use $(k_1, k_2) \Rightarrow X^{k_2} Z^{k_1}$.
- For qubit 2 use $(k_3, k_4) \Rightarrow X^{k_4} Z^{k_3}$.

If $T_A = (1, 0, 0, 1)$ then:

$$|S\rangle = \left(Z^1 X^0 |p'_1\rangle\right) \otimes \left(Z^0 X^1 |p'_2\rangle\right) = (Z\,|p'_1\rangle) \otimes (X\,|p'_2\rangle).$$

During decryption, SKG applies again $X^0 Z^1$ on qubit 1 and $X^1 Z^0$ on qubit 2, restoring $|p'_1\rangle \otimes |p'_2\rangle$, then applies $U^\dagger$ to recover $|P\rangle$.

7 Quantum Key Distribution (QKD) for Secure Transmission

To ensure secure transmission of CO_2 sensor data to the SKG, Quantum Key Distribution (QKD) protocols can be used. QKD allows two parties, Alice (sensor) and SKG, to establish a shared secret key using quantum properties. This key is later used to encrypt the sensor data with the Quantum One-Time Pad (QOTP).

7.1 Most Suitable QKD Protocols

7.1.1 BB84 (Bennett & Brassard, 1984)

- Alice sends qubits in two random bases:
  1. Computational basis: $\{|0\rangle, |1\rangle\}$
  2. Diagonal basis: $\{|+\rangle, |-\rangle\}$, where $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$
- SKG measures each qubit in a random basis.
- Alice and SKG publicly compare bases and keep only bits measured in the same basis to form the secret key.
- A fraction of bits is used to test for eavesdroppers (Eve).

Advantages: Simple, experimentally tested, secure even against quantum attackers. **Limitations:** Requires quantum hardware or single-photon sources.

7.1.2 Decoy-state BB84

- Variant of BB84 that sends "decoy" signals along with real ones to detect multi-photon attacks.
- Increases robustness for practical fiber-optic implementations.

7.1.3 Continuous Variable (CV) QKD

- Uses quadratures of the electromagnetic field (amplitude/phase) instead of single photons.
- Can be detected with classical detectors (homodyne), simplifying sensor integration.
- Less noise-tolerant, suitable for short-to-medium distances.

7.2 Recommendation

For CO_2 sensors transmitting data to an SKG, practical options are:

1. Standard BB84 or Decoy-state BB84 for solid, proven security.
2. CV-QKD if simpler integration with sensors without single-photon hardware is needed.

The key obtained via QKD is subsequently used to encrypt sensor data with the Quantum One-Time Pad (QOTP), ensuring confidentiality and authenticity.