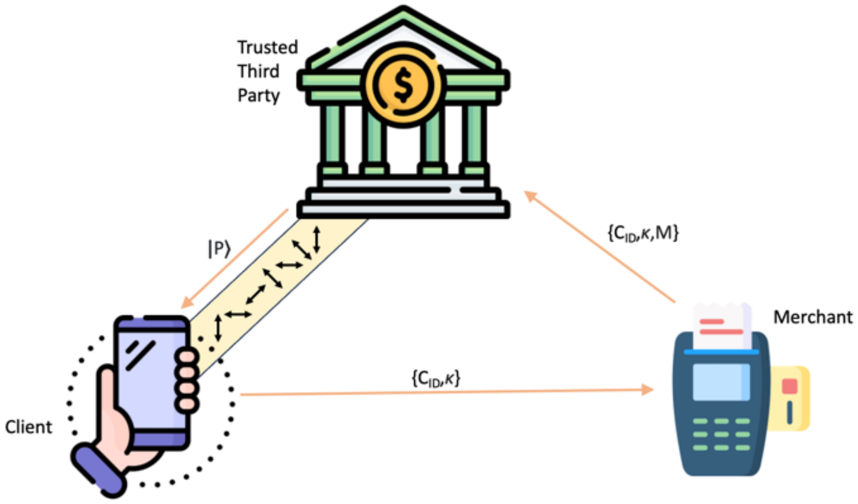
Secure Quantum Digital Payments

The team should design and implement, using SquidASM, a simulation of the quantum-digital payment protocol described in paper [1].

There are three parties: a Client, a Merchant, and a Bank/Creditcard institute (denoted as Trusted Third Party, TTP). We do not assume any quantum or classical communication channel to be trusted, except an initial prior step between the TTP and Client for an account creation (in which the Client receives a secret token C from the TTP).



The protocol:

1. During a payment, the TTP generates a random bitstring b and a random conjugate basis-string B, both of length λ. The j-th bit of b is encoded onto a quantum state prepared in the j-th element of B. For example, assuming λ = 4 and Bj in {+/-;0/1}, choosing b = 0101 and B = 0011 would result in a 4-qubit quantum state |P⟩ = |+⟩|-⟩|0⟩|1⟩. Indeed, the first bit of B (i.e., 0) selects +/- and the first bit of b (i.e., 0) selects +. And so on. The TTP sends |P⟩ to the Client.
2. Then, the Client calculates m=MAC(C,M), denoted as output tag, where M is the identifier of the Merchant and MAC is a Message Authentication Code. The client interprets m as a basis-string and privately measures |P⟩ according to m. The resulting string of measurements, denoted as κ, constitutes a cryptogram.
3. The Client sends its public identifier C_{ID} (not to be confused with C, which is a secret between the Client and the TTP) to the Merchant, along with κ. The Merchant sends {C_{ID},κ,M} to the TTP for verification.
4. To authorize the purchase, the TTP looks up C (the secret token corresponding to C_{ID}) and calculates m = MAC(C,M). The TTP accepts the transaction (and transfers the money from the Client's account to the Merchant's one) if and only if $\kappa_j = b_j$ when $m_j = B_j$. The TTP rejects otherwise.

It is suggested to start by simulating the execution of the protocol described above, assuming that the Client and the Merchant are both honest. For simplicity, assume that C is already shared between the Client and the TTP, when the protocol execution starts.

Then, also simulate malicious behaviours.

- A malicious merchant M’ would try to forge an output tag such that MAC(C,M)=MAC(C,M’) ⇔ m=m’ ⇔ $\kappa=\kappa'$. In this way, merchant M’ could receive the payment that should be sent to merchant M. The value of parameter λ should be selected so that the probability of output tag forging is minimised (see [1] for details). Show that non-ideal choices for λ may lead to wrong payments with high probability.
- A malicious Client would try to exploit the imperfection of real devices (inaccurate state preparation, lossy quantum channels, etc.) to circumvent the commitment or double-spend the cryptogram. In fact, some bits in step 4 will be unequal, although measured in the same basis, and the protocol would abort even though it was followed honestly. Suppose the TTP tolerates as many as 50% losses. A malicious Client could measure half of the quantum token |P⟩ in the basis for M_0 and the other half in the basis for M_1 (being M_0 and M_1 two merchants), thus creating two successfully committed tokens. Try to figure out how to prevent this kind of attack (a possible solution is illustrated in [1]).

The team should feel free to choose the preferred qubit technology, among those available with SquidASM (generic hardware, NV centers, color centers). Plot the fidelity of the quantum states for different values of the physical parameters for qubits and quantum links.

References

[1] Peter Schiansky, Julia Kalb, Esther Sztatecsny, Marie-Christine Roehsner, Tobias Guggemos, Alessandro Trenti, Mathieu Bozzio, and Philip Walther. Demonstration of quantum-digital payments. Nature Communications, 14(1), Jun 2023.