Master internship: Zero-Knowledge proofs and applications (threshold signatures, confidential transactions, secure data management etc.)

Pierre-Alain Fouque*, Matthieu Rambaud†, and Weiqiang Wen†

*IRISA Rennes, †IUF, †@telecom-paris.fr

Threshold cryptography are mechanisms enabling a number $n$ distrustful participants to obtain the output of some computation on secret data as soon as a threshold number of them participates, but reveals nothing otherwise. The most used one in the industry, called threshold signatures, enables $n$ machines to produce a single signature on behalf of a client. Even if the adversary corrupts a threshold number of machines ($t < n$) and learns their shares of the client’s signing key, it will not be able to forge a valid signature. This technique is used by Coinbase (32Bn$ market cap) to manage the crypto-wallets of more than 5M clients [Coi23], and also by Fireblocks (8Bn$ market cap). Even this simple example uses a number of ingredients, each of them having many other applications. We list some of them, which have been recently called for standardization by the NIST (national institute of standards and technology) [NIS23b]. First, the pieces of the key are frequently re-sampled afresh (after every signing at Coinbase [Lin23], every minutes at Fireblocks [Fir21]), so that the old pieces are somehow useless to the adversary, but counter-intuitively the new pieces enable to sign under the same key. This provides a so-called proactive security level against a mobile adversary, which is encouraged by the NIST. Counter-intuitively, the key never appears in the clear, its pieces are instead collectively generated by the machines (DKG) [NIS23a], which eliminates any single point of failure. Counter-intuitively, each machine is able to prove to the others that it correctly generates [Lin23] and uses its piece of the key, without disclosing it.

The tool central to all the previous ingredients are called zero-knowledge proofs (NIZKs), they are used everywhere in threshold cryptography and blockchains. For example, NIZKs enable a large subset of users [Dai23] to prove collectively that they executed valid transactions with each other, without revealing the prices (the keywords are “ZK-rollups”, “layers 2”). It is even possible to keep the prices confidential within this subset: the French startup Zama [Zam23] has issued a threshold (“fully homomorphic”) mechanism, adopted by some layers 2, enabling a group of $n$ machines to match bids and offers without even seeing them. NIZKs of correct threshold decryption enable these machines to prove that the final price, which they collectively decrypt, is the correct one.

The goal of the internship is to tailor-make and simplify, for specific applications, NIZKs based on mathematical problems assumed resilient to quantum computers. Our favorite choice are NIZKs based on hard lattices problems [AL21; LNP22], since they have an easy integration into threshold decryption and signatures. Furthermore, they can be optimized with fancy algebra [CLM23; BF22]. However, depending on the candidate’s taste, other applications can be considered, with possibly different NIZK systems, e.g. the popular and funny combination of decoding and “MPC-in-the-head” [AGH+23].

References


