**Summary of the MPC project**

**1 Background**

In digital assets, in order to make any transaction happen, we need two keys, a public key, and a private key. Public key is visible to everyone and private key must only be known to the owner. Private key management is extremely important because if your private key is accessible then your account’s data can be manipulated.

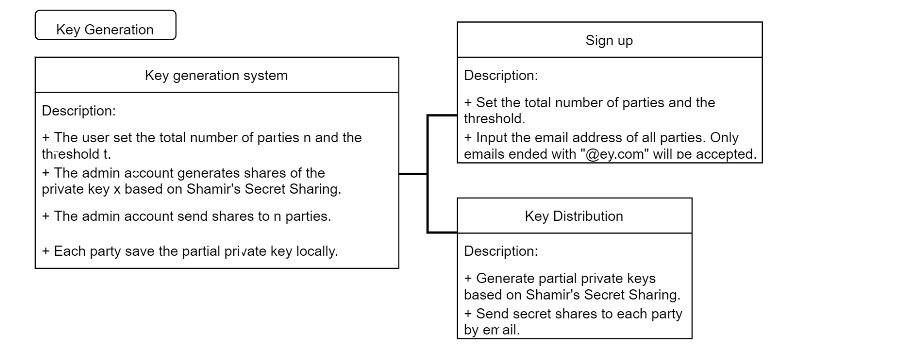
Multi-party computation (MPC) is a subfield of cryptography that enables multiple parties to jointly compute a function without revealing the inputs. Threshold Signature Scheme (TSS) is a subfield in MPC. TSS offers highly efficient distributed private key and signature generation. In TSS, the private key is distributed into n parties, and a subset of size t (t < n) are involved in the computation of the signature for the transaction. This makes digital assets safe, because never ever any server can get access to the entire private key, even if the attacker gets access to one or two servers cannot get the entire private key.

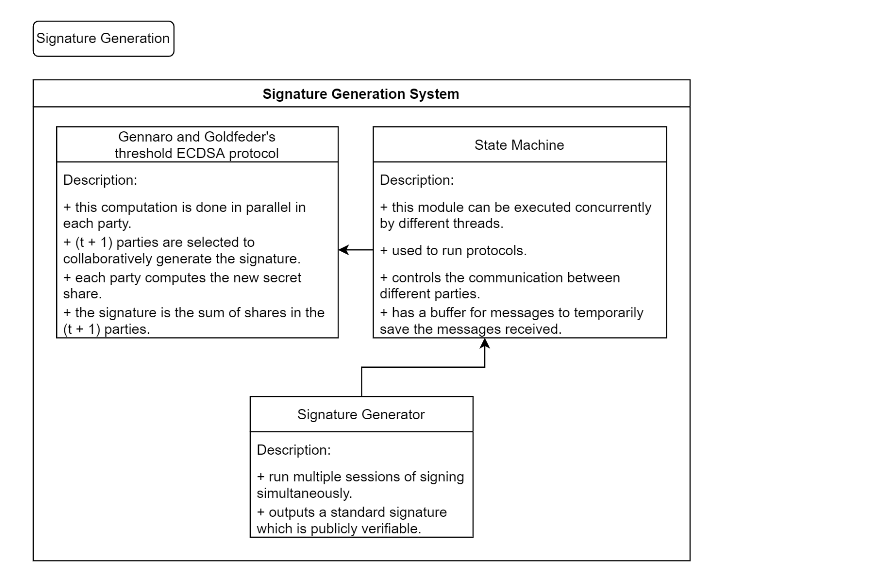
**2 Goal**

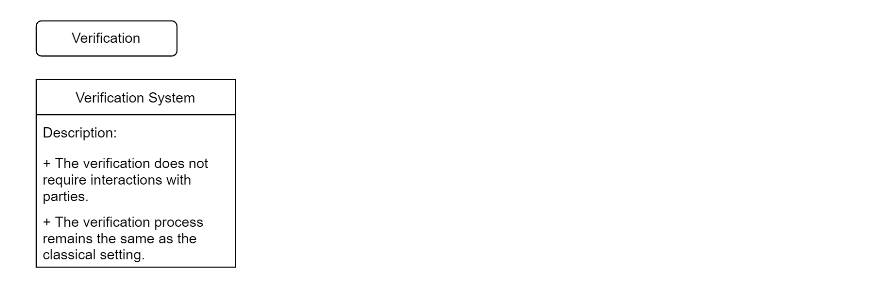
The goal of the project is mainly applying MPC (multi-party computation) to the current QAS coin platform to secure the private key generation and management. The Threshold Signature Scheme based on Shamir’s secret sharing will be implemented. The outcome should be a MPC backend which supports distributed key generation, signature generation and verification. The MPC backend should also match the input and output format of the current QAS coin platform, making future combination easier.

2.1 Flow chart

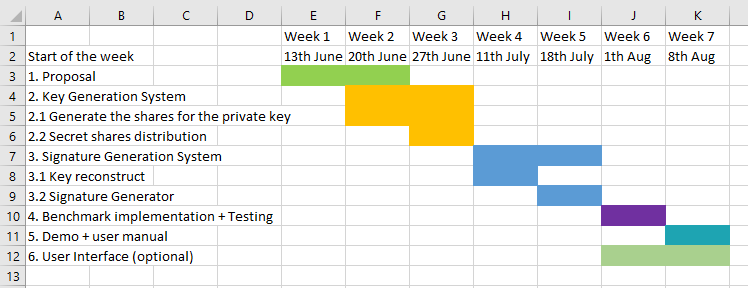
The components to implement are in the diagram below. This flow chart shows the basic design of the system. There are mainly three parts, key generation system, signature generation system and verification system.







2.2 Timeline



**3 Status**

The implementation of the key generation system is done. The Shamir’s secret sharing algorithm is implemented in RUST. The code is uploaded to the github repository <https://github.com/VivianH61/MPC_QAScoin/tree/main/src>.

3.1 Logic of the key generation system

The key generation is based on Shamir’s (K, N) threshold scheme. S is the secret that we wish to encode. It is divided into N parts: S1, S2, S3, …., Sn. After dividing it, a number K is chosen by the user in order to decrypt the parts and find the original secret. If we know less than K parts, then we will not be able to find the secret S (i.e.) the secret S cannot be reconstructed with (K – 1) parts or fewer.

3.2 Mathematical principle

Interpolation theorem: There exists a unique polynomial of degree at most k - 1 that interpolates the k data points. The idea is to build a polynomial with the degree (K – 1) such that the constant term is the secret number and the remaining (K – 1) coefficients are random. This constant term can be found by using any K points out of N points generated from this polynomial.

Example: Let the secret code S = 65, N = 4, K = 2.

1. Initially, in order to encrypt the secret code, we build a polynomial of degree (K – 1).
2. Therefore, let the polynomial be y = a + bx. Here, the constant part ‘a’ is our secret code.
3. Let b be any random number, say b = 15.
4. Therefore, for this polynomial y = 65 + 15x, we generate N = 4 points from it.
5. Let those 4 points be (1, 80), (2, 95), (3, 110), (4, 125). Clearly, we can generate the initial polynomial from any two of these 4 points and in the resulting polynomial, the constant term a is the required secret code.

3.3 Demo

The current key generation system follows the logic of the Shamir’s secret sharing algorithm. The private key is randomly generated with the format of 64 hexadecimal characters, which matches the format of the private key for Ethereum. The user can input the total number of shares N, and the threshold K in the terminal. In the example below, the private key as the secret is distributed into 10 shares, and the entire private key can be reconstructed by any 3 shares. The 10 shares generated from the private key are printed in the terminal.



