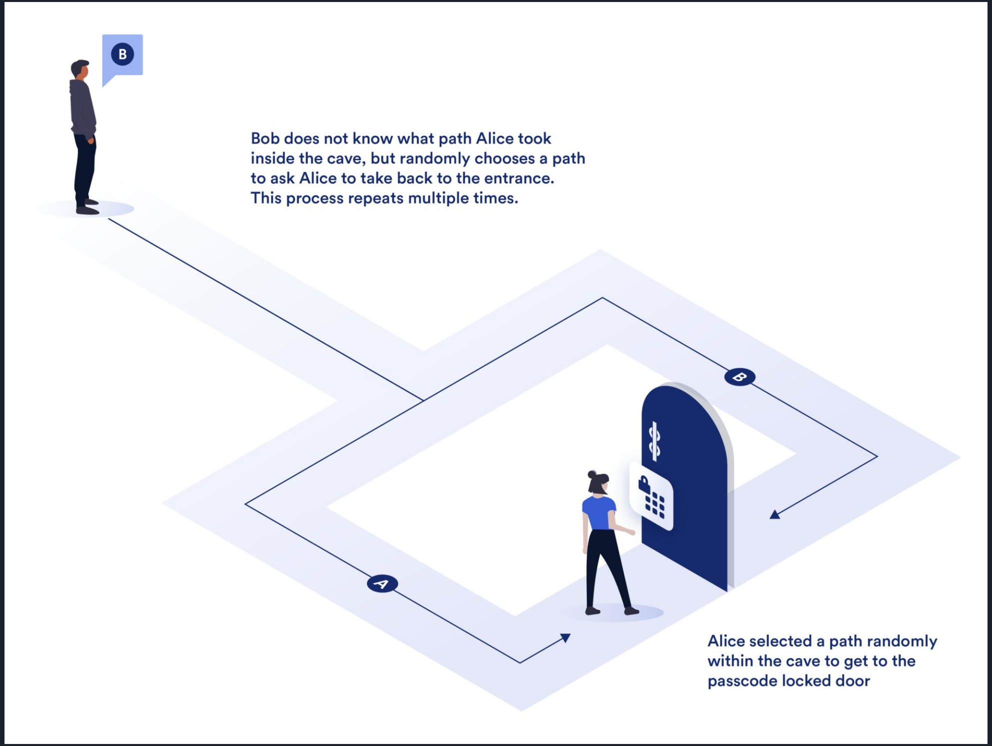
Chapter 3 - Signature

Section 3.1 - Σ-protocols and the Fiat-Shamir Heuristic

Interactive Proof of Knowledge

Zero-knowledge

Legitimate prover knows (x, w)



Endomorphism Ring

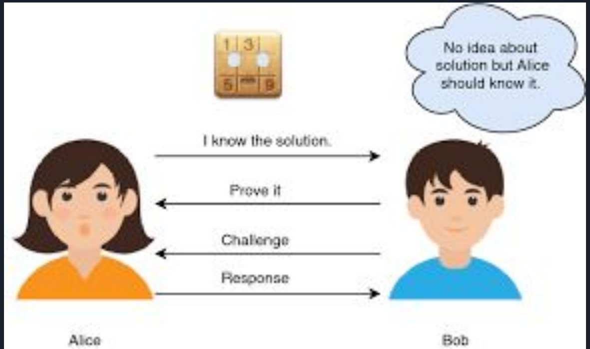
An endomorphism given an elliptic curve is an isogeny (mapping), $\varphi : E \rightarrow E$

The collection of all endomorphisms along with addition and noncommutative multiplication for the endomorphism ring of an elliptic curve

$(f+g)(P) = f(P) + g(P)$

$(f \circ g)(P) = f(g(P))$

Three interactive phases



Public key: E_A

Private key (knowledge): $End(E_A)$

1) Commitment: Prover randomly generates $(E_1, End(E_1))$. Sends E_1 to verifier.
2) Challenge: Verifier randomly generates $\varphi_{chall} : E_1 \rightarrow E_2$ and sends φ_{chall} to prover.
3) Response: Prover uses $End(E_1)$ and φ_{chall} to compute $End(E_2)$. Then uses this and its knowledge to compute $\varphi_{resp} : E_A \rightarrow E_2$. Sends this to Verifier

Verifier who has the public key and E_2 can easily check if φ_{resp} is a correct isogeny

There are some complications but this is the main idea

Fiat-Shamir Transform

Interactive to non-interactive proof of knowledge

Single signing and single verifying stage without explicit communication

Key difference is prover computes its own challenge using an unpredictable hash function that changes drastically for different commitments or different messages

Section 3.3 - Key Generation

SQIsign.KeyGen Algorithm

Input: 1^λ where λ is the security parameter
Output: Secret signing key $\mathtt{sk}$ and public verification key $\mathtt{pk}$
Output: $\mathtt{found}$ a boolean indicating whether computation succeeded

Select a random $\mathtt{KLPT\_secret\_key\_prime\_size}$-bit prime $D_{\text{secret}} \equiv 3 \bmod 4$

An element is chosen $\gamma \in \mathcal{O}_0$ using the FullRepresentInteger algorithm. Also a random positive $a < D_{secret}$ is chosen. Then a secret ideal is computed: $I_{secret} = \mathcal{O}_0 (\gamma(a + i) + \mathcal{O}_0(D_{secret})$

$\alpha, \mathtt{found} := \mathsf{KeyGenKLPT}_{2^\bullet}(I_{\text{secret}})$

KeyGenKLPT returns a quaternion that will be used to connect I_{secret} to an equivalent ideal (a multiple of I_{secret}) with a different norm

Section 3.3 - Key Generation

$$\chi_I(\alpha) = I\frac{\bar{\alpha}}{\mathrm{nrd}(I)}$$

$$J_{\text{secret}} := \chi_{I_{\text{secret}}}(\alpha)$$

J_{secret} is that equivalent ideal with power-of-2 norm

$$\varphi_{\text{secret}}, \_, \texttt{found} := \mathsf{IdealToIsogenyEichler}_{2^\bullet}(J_{\text{secret}}, \mathcal{O}_0, B_{0,T})$$

$$E_0 : y^2 = x^3 + x$$

$B_{0,T}$ is a basis for $E_0[T]$, the T-torsion subgroup of E_0 (all points P_0 on E_0 such that $[T]P_0 = \infty$)

This is efficiently calculated (because T is smooth) by an algorithm similar to Pohlig-Hellman

Section 3.3 - Key Generation

$$E_A, \varphi_{\text{secret}} := \mathsf{Normalized}(\varphi_{\text{secret}})$$

The public key, E_A, is found using the Normalized algorithm and φ_{secret} is updated to map to this curve

$$B_{A,T} := \varphi_{\text{secret}}(B_{0,T})$$

φ_{secret} is applied to find a basis for the T-torsion subgroup of E_A

Let P be a point generating $\ker \varphi_{\text{secret}} \cap E_0[2^f]$
$(P, Q) := \mathsf{CompleteBasis}_{2^f, p+1}(E_0, P)$

P is one of the basis points of the 2^f-torsion subgroup of E_0. The other one, Q, is found using the CompleteBasis algorithm

Section 3.3 - Key Generation

$$Q := \varphi_{\text{secret}}(Q)$$

The generating point Q on E_0 is mapped to its image on E_A

Set $\mathrm{pk} := E_A$
Set $\mathtt{sk} := (\alpha, B_{A,T}, Q)$
end if
return $\mathtt{sk}, \mathrm{pk}, \mathtt{found}$

The signing key (the knowledge) is the connecting quaternion, T-torsion subgroup basis, and mapped basis point Q