

1 The trapdoor

It works as follows, $A, B \in \mathcal{M}_b$ will be scalar matrices.

Public parameters: E_0, d_1, d_2 .

Trapdoor parameters (public key, secret key): $(pk, sk) = (E_A, (A, \phi_A))$.

Input of the trapdoor: $K_1 \subset E_0[d_1], K_2 \subset E_A[d_2], B$.

Evaluation:

$$\begin{array}{ccccccc}
 & & K_1 & & K_2 & & \\
 & & \cap & & \cap & & \\
 \begin{pmatrix} P_b \\ Q_b \end{pmatrix} & \in & E_0 & & E_A & \ni & A * \begin{pmatrix} \phi_A(P_b) \\ \phi_A(Q_b) \end{pmatrix} \\
 & & \downarrow \phi_1 & & \downarrow \phi_2 & & \\
 \begin{pmatrix} R_1 \\ S_1 \end{pmatrix} & = & B * \begin{pmatrix} \phi_1(P_b) \\ \phi_1(Q_b) \end{pmatrix} & \in & E_1 & & E_2 & \ni & B * A * \begin{pmatrix} \phi_2 \circ \phi_A(P_b) \\ \phi_2 \circ \phi_A(Q_b) \end{pmatrix} & = & \begin{pmatrix} R_2 \\ S_2 \end{pmatrix}
 \end{array}$$

Output: $(E_1, R_1, S_1, E_2, R_2, S_2)$.

Now to inverse the map, we have access to:

- the secret key $(A, \phi_A : E_0 \rightarrow E_A)$
- the image points

$$\begin{pmatrix} R_1 \\ S_1 \end{pmatrix} = B * \begin{pmatrix} \phi_1(P_b) \\ \phi_1(Q_b) \end{pmatrix}$$

- and the image points

$$\begin{pmatrix} R_2 \\ S_2 \end{pmatrix} = B * A * \begin{pmatrix} \phi_2 \circ \phi_A(P_b) \\ \phi_2 \circ \phi_A(Q_b) \end{pmatrix}$$

Consider $\psi = \phi_2 \circ \phi_A \circ \widehat{\phi_1}$, which has degree $= d_2 d_A d_1$, we have

$$B * A * B^{-1} * \begin{pmatrix} \psi(R_1) \\ \psi(S_1) \end{pmatrix} = [d_1] \begin{pmatrix} R_2 \\ S_2 \end{pmatrix}$$

But A and B commute so that since we know A and d_1 we can recover $\begin{pmatrix} \psi(R_1) \\ \psi(S_1) \end{pmatrix}$. Now we would like to recover ψ . We have the torsion point images of order $2^b > d_1 d_A d_2$ so that we can apply the usual torsion attacks. The parameters just need to be worked out.

It happens that, under the CIST² assumption, the FESTA trapdoor verifies the following definition:

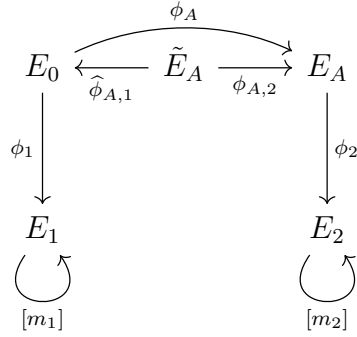
Definition 1.0.1 (Quantum partial domain one-way function). Let X_0 , X_1 and Y be three finite sets. A function $f : X_0 \times X_1 \rightarrow Y$ is a quantum partial-domain one-way function if, for any polynomial-time quantum adversary A , the following holds:

$$P(s' = s | s \leftarrow X_0, t \leftarrow X_1, s' \leftarrow A(f(s, t)))$$

Then the OAEP transform, as described [here](#) builds a PKE from such a trapdoor functions.

2 The concrete instantiation

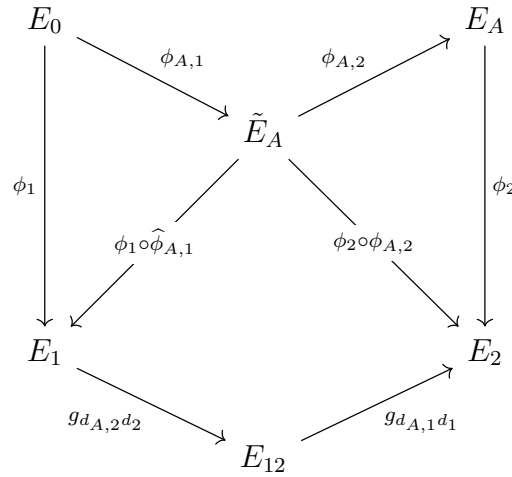
Consider the diagram:



Where we decomposed ϕ_A in $\phi_{A,1} \circ \phi_{A,2}$. The main idea here would be to directly get

$$\phi_2 \circ \phi_A \circ \hat{\phi}_1$$

from the the 2^b torsion point images. We would need to have $2^b - d_2 * d_A * d_1$ smooth. Which gives few choices and usually low efficiency. Instead, since we already have ϕ_A to invert the trapdoor. The idea would be to decompose ϕ_A as $\phi_{A,1} \circ \phi_{A,2}$ and use the hidden diagram



In which finding good parameters amounts to solving

$$m_1^2 d_{A,1} d_1 + m_2^2 d_{A,2} d_2 = 2^b$$

The trick of decomposing with some scalar multiplication doesn't change anything to the security! The paper proposes a way to find solutions with the desired properties efficiently. The d'_i s are all squares.