Introduction to Pairings
Diego F. Aranha
dfaranha@ic.unicamp.br

Please check the slides (or Wikipedia) for the protocol specifications!

IBE with Type-1 Pairing

Implement the Boneh-Franklin Identity-Based Encryption (slide 20) system using the Weil pairing defined over a supersingular curve. You can follow the steps below to define a toy curve we can play with (thanks to Craig for the parameters!):

1. Let \( p = 7691 \) and let \( E/\mathbb{F}_p : y^2 = x^3 + 1 \) with embedding degree \( k = 2 \). Define \( \mathbb{F}_{p^2} = \mathbb{F}_p(i) \) where \( i^2 + 1 = 0 \). Let \( P = (2693, 4312) \in E(\mathbb{F}_p) \). Both curves \( E(\mathbb{F}_p) \) and \( E(\mathbb{F}_{p^2}) \) have a subgroup of points or order \( r = 641 \). Define the base field, quadratic extension and elliptic curve. Compute and factor the order of the curve over the base field and quadratic extension.

2. Implement a function to hash strings to points in the curve. A simple (and non-ideal) deterministic way of performing this consists in hashing to the \( x \)-coordinate of the point and incrementing \( x \) until a point is found. You can use the function \texttt{SHA1} in MAGMA that receives as input a string of hexadecimal digits and function \texttt{StringToInteger} to convert the result back to an integer. Beware of cofactors!

3. Implement a function to hash elements from \( \mathbb{F}_{p^2} \) to integers. This can be done by splitting the input into two \( \mathbb{F}_p \) elements, converting the concatenation to a string and using \texttt{SHA1}.

4. Implement the key generation, encryption and decryption functions for the scheme. For a suitable pairing, you can use the \texttt{WeilPairing} in MAGMA which receives points over \( E(\mathbb{F}_{p^2}) \). A distortion map \( \psi : (x, y) \to (\xi_3 x, y) \) is defined for this curve for cube root \( \xi_3 \). The \( \oplus \) operator can be implemented through \texttt{BitwiseXor} which receives integers as operands.

5. Verify the correctness of the implementation using a test message.

\textbf{Important:} Please notice that the \texttt{SHA1} hash function is insecure and was used for illustration purposes only (and because MAGMA apparently does not have \texttt{SHA2}).
BLS with Type-1 and Type-3 Pairings

Implement the BLS signature scheme (slide 17) using parameters from the previous section.

**Bonus:** Start a new implementation instantiating the pairing groups with prime-order Barreto-Naehrig curves having embedding degree \( k = 12 \). You can check the last few slides for some background material on BN curves, as specified below:

1. Let \( x = -(2^{62} + 2^{55} + 1) \). Let \( p(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1 \) and \( r(x) = 36x^4 + 36x^3 + 18x^2 + 6x + 1 \) define primes \( p, r \) for the base field and the curve order, respectively. Define the extensions \( \mathbb{F}_{p^2}, \mathbb{F}_{p^6}, \mathbb{F}_{p^{12}} \) as in the slides using the quadratic/cubic non-residue \( \xi = i + 1 \) for \( \mathbb{F}_{p^2} = \mathbb{F}_p(i) \) and \( i^2 + 1 = 0 \).

2. Define the curve over the base field \( E(\mathbb{F}_p) : y^2 = x^3 + 2 \) and the sextic \((d = 6)\) twist \( E(\mathbb{F}_{p^2}) : y^2 = x^3 + 2/\xi \). Compact generators for \( E(\mathbb{F}_p) \) and \( E(\mathbb{F}_{p^2}) \) are \( G = (-1, 1) \) and \([h]G' \) for cofactor \( h = 2p - r \) and \( G' = (-i, 1) \). This is a set of realistic parameters initially proposed at the 128-bit security level, but it is now around 100 bits due to the recent attacks.

3. For the choice of pairing, use the `ReducedTatePairing` from MAGMA, which takes points with coordinates over the full extension \( \mathbb{F}_{p^{12}} \). The untwisting isomorphism maps points from \( E(\mathbb{F}_{p^2}) \) to \( E(\mathbb{F}_{p^{12}}) \) by computing \( \psi : (x, y) \rightarrow (\xi^2x, \xi^3y) \).

4. Implement the key generation, signature and verification functionalities. Choose wisely which groups among \( \mathbb{G}_1 \) and \( \mathbb{G}_2 \) will represent public keys and signatures.

5. Verify the correctness of the scheme by signing and verifying a simple message.