Advanced Crypto – Exercises

I. Security assumptions for pairing-based cryptography

Let G_1, G_2, G_3 be three cyclic groups of prime order ℓ and $e : G_1 \times G_2 \to G_3$ a bilinear, non-degenerate map. Even though G_1 and G_2 are usually subgroups of an elliptic curve, we will use multiplicative notations for the three groups.

We recall some standard problems:

- Discrete logarithm problem in G_i (DLP_i): given g and g^a in G_i, find a.
- Computational Diffie-Hellman problem in G_i ($CDHP_i$): given g, g^a and g^b in G_i, compute g^{ab}.
- Decisional Diffie-Hellman problem in G_i ($DDHP_i$): given g, g^a, g^b and g^c in G_i, determine if $ab = c$.
- Co-computational Diffie-Hellman problem (*Co-CDHP*): given g, g^a in G_1 and h in G_2, compute h^a.
- Co-decisional Diffie-Hellman problem (*Co-DDHP*): given g, g^a in G_1 and h, h^b in G_2, determine if $a = b$.

1. Give all the reductions that do not use pairings between the above problems.
2. Using the pairing $e : G_1 \times G_2 \to G_3$, which of the above problems become easy? What are the additional reductions?
3. Same question, assuming there exists an efficiently computable homomorphism $\psi : G_2 \to G_1$ (this is sometimes called a Type 2 pairing).

A natural question is whether the computation of the pairing is reversible. This is captured by the following problems:

- Inversion problem on G_1 (INV_1): given $h \in G_2$ and $\mu \in G_3$, find $g \in G_1$ such that $e(g, h) = \mu$
- Inversion problem on G_2 (INV_2): given $g \in G_1$ and $\mu \in G_3$, find $h \in G_2$ such that $e(g, h) = \mu$
- Inversion problem (*INV*): given $\mu \in G_3$, find $(g, h) \in G_1 \times G_2$ such that $e(g, h) = \mu$.

4. Show that if INV_2 is easy, then *Co-CDHP* is easy as well.
5. Show that if INV_1 and INV_2 are easy, then $CDHP_3$ is easy as well.
6. Is there any obvious reduction from *INV*?

The security of the identity-based encryption system of Boneh and Franklin relies on the hardness of yet another problem:

- (Co-)Bilinear Diffie-Hellman problem (*BDHP*): given $g, g^a, g^b \in G_1$ and $h \in G_2$, compute $e(g, h)^{ab}$.

7. Show that *BDHP* is easy if any one of $CDHP_1$, $CDHP_3$, *Co-CDHP*, INV_2 is easy.

II. Batched Boneh-Lynn-Sacham signatures

We recall the BLS signature scheme:

- Setup: fix a pairing $e : G_1 \times G_2 \to G_3$ between three groups of order p in which the discrete logarithm problem is hard, and such that the co-computational Diffie-Hellman problem is also hard. Fix a generator g of G_1, as well as a cryptographic hash function $H : \{0,1\}^* \to G_2$.
- Key generation: the secret key of a user is an integer $s \in \mathbb{Z}/p\mathbb{Z}$, and its public key is $h = g^s$.
- Signing: the signature of a message m is $\sigma = H(m)^s$.
- Verification: in order to verify that σ is indeed the signature of a message m by a user whose public key is h, check that $e(g,\sigma) \stackrel{?}{=} e(h, H(m))$.

1. Recall why this protocol is correct and secure.

Assume now that n users, with public keys $h_1, \ldots, h_n$, send n signed messages (m_i, σ_i). We can now define the batched protocol:

- Aggregate signatures: the batched signature of the messages is $\sigma_{tot} = \prod_{i=1}^n \sigma_i$. This can be computed by anyone, for instance by an e-mail provider.
- Verification: in order to verify that the batched signature σ_{tot} is correct, compute $v_i = e(h_i, H(m_i))$ for all $1 \le i \le n$ and check whether $e(g, \sigma_{tot}) \stackrel{?}{=} \displaystyle\prod_{i=1}^n v_i$.

2. Show that the final test is indeed an equality if all the signatures are correct. What is the interest of this batched verification? Compare to the cost of n separate verifications.
3. If all the messages come from the same user (i.e. $h_1 = \cdots = h_n$), explain how to further simplify the verification.
4. The rogue public-key attack.
   Let h be the public key of a legitimate user Alice. An attacker can register the value $h' = g^r h^{-1}$ as his own public key (where r is random), and pretend that Alice and himself have sent the same message m, by presenting the aggregate signature $\sigma_{tot} = H(m)^r$ (or in fact, any two signatures σ, σ' such that $\sigma\sigma' = H(m)^r$).
   (a) Show that this passes the verification test, and thus convinces that Alice has sent the message m.
   (b) Propose a simple counter-measure to this attack.

III. Hashing to an elliptic curve

Several protocols, and notably BLS short signatures, require a cryptographic hash function $H : \{0;1\}^* \to E(\mathbb{F}_q)$ (or a subsgroup G of $E(\mathbb{F}_q)$). This is not as easy as it may seem.

1. A deeply flawed solution.
   Let P be a generator of G (of large prime order), and h a standard cryptographic hash function that outputs n bits; these n bits can be interpreted as an integer in $[\![0; 2^n - 1]\!]$. A first idea is to define a function H by:
   $$\forall m \in \{0;1\}^*,\ H(m) = h(m)P$$
   Show that this completely breaks the BLS protocol. More precisely, show that an attacker who knows a single BLS-signed message (m, σ) of a legitimate user can then forge the signature of any message.
2. Two less flawed solutions.
   Let $E : y^2 = x^3 + ax + b$ an elliptic curve defined over $\mathbb{F}_q$, and h a standard cryptographic h function as above. We propose two methods to hash a message m into E.

- First method: compute $h(m)$ and interpret the result (save one bit) as an element of $\mathbb{F}_p$. If is is the x-coordinate of an element of E (i.e. if $h(m)^3 + ah(m) + b$ is a square), ouput the corresponding point, using the saved bit to decide between the two possible values of the y-coordinate. Otherwise, increment $h(m)$ until finding the x-coordinate of a point.
- Second method: set $i = 0$, compute $h(m||i)$ and interpret the result (save one bit) as an element of $\mathbb{F}_p$. If is is the x-coordinate of an element of E (i.e. if $h(m)^3 + ah(m) + b$ is a square), ouput the corresponding point, using the saved bit to decide between the two possible values of the y-coordinate. Otherwise, increment i until finding the x-coordinate of a point.

What are the pros and the cons of these two methods? Why are they not satisfactory?

3. A particular case.
Let p be a prime number such that $p = 2 \bmod 3$ and E the elliptic curve over $\mathbb{F}_p$ with equation $y^2 = x^3 + b$. Let h a standard cryptographic h function as above. Interpreting the output of h as an element of $\mathbb{F}_p$, we define:
$$\forall m \in \{0;1\}^*,\ H(m) = \left(\left(h(m)^2 - b\right)^{\frac{2p-1}{3}}, h(m)\right)$$
   (a) Show that this is indeed a point of E.
   (b) Prove that $\#E(\mathbb{F}_p) = p + 1$ (i.e. E is *supersingular*) and compute its embedding degree.
   (c) What are the pros and the cons of this method?