Problem 1: Malleability of ElGamal

Remember the auction example from the lecture: Bidder 1 produces a ciphertext $c = E(pk, bid_1)$ where $E$ is the ElGamal encryption algorithm (using integers mod $p$ as the underlying group). Given $c$, Bidder 2 can then compute $c'$ such that $c'$ decrypts to $2 \cdot bid_1 \mod p$. This allows Bidder 2 to consistently bid twice as much as Bidder 1.

Now refine the attack. You may assume that $bid_1$ is the amount of Cents Bidder 1 is willing to pay. And you can assume that Bidder 1 will always bid a whole number of Euros. (I.e., $bid_1$ is a multiple of 100.)

Show how Bidder 2 can consistently overbid Bidder 1 by only 1%. What happens to your attack if Bidder 1 suddenly does not bid a whole number of Euros?

**Hint:** Remember that modulo $p$, one can efficiently find inverses. For example, one can find a number $a$ such that $100a \equiv 1 \mod p$.

**Solution.** Let $a$ be the inverse of 100 modulo $p$. That is $100a \equiv 1 \mod p$. If Bidder 2 intercepts a message $c = E(pk, bid_1)$, he computes $c' = E(pk, (1 + a)bid_1 \mod p)$. This way he bids to $bid_2 := (1 + a)bid_1 \mod p$. We have to show that $bid_2 = 1.01 \cdot bid_1$.

We assumed that Bidder 1 always pays a whole amount of Euros. Hence $bid_1 = 100x$ for some integer $x$. Thus $1.01bid_1 = 101x$. Furthermore,

$$bid_2 \equiv (1 + a)bid_1 = (1 + a)100x \equiv 100x + (100a)x \equiv 100x + x \equiv 101x \mod p.$$ 

Thus we know that $bid_2 \equiv 101x \mod p$. Furthermore, since $p$ is a large prime, it is to be expected that $101x < p$ (otherwise Bidder 1 would bid an unrealistically large amount). Thus $bid_2 = 101x$. Hence $bid_2 = 1.01bid_1$.

What happens if Bidder 1 does not bid a whole amount of Euros? For example, if $bid_1 = 1$ (1 Cent). In this case, Bidder 2 will bid $1 + a$ Cent. And $a$ is a very large number. (At least $a > p/100$ because otherwise $100a < p$ and hence $100a \not\equiv 1 \mod p$.)

Thus in this case, Bidder 2 may end up bidding huge amounts. (Typically in the order of magnitude of $p$.)

---

1As long as $bid_1 < p/2$, that is. Otherwise $2 \cdot bid_1 \mod p$ will not be twice as much as $bid_1$. However, for large $p$, $bid_1 \geq p/2$ is an unrealistically high bid.
**Problem 2: An alternative hash construction**

Consider the following hash function on messages the length of which is a multiple of 128. We fix a secure block-cipher $E$ (a strong pseudorandom permutation) with block length 128. We randomly pick a key $k$ for the scheme. The cipher $E$ and the key $k$ are publicly known. To hash a message $m$, we partition it into blocks $m_1, \ldots, m_t$ each of which has length 128. $i.e$ $m = m_1||m_2|| \ldots ||m_t$. We define 

$$H(m) := E_k(m_1) \oplus E_k(m_2) \oplus \ldots \oplus E_k(m_t).$$

1. Is $H$ collision-resistant? If yes, explain why. If not, show an attack against the scheme. (The adversary is supposed to find a pair $x, x'$ such that $x \neq x'$ and $H(x) = H(x')$.)

2. Is $H$ preimage resistant? If yes, explain why. If not, show an attack against the scheme. (The adversary is given a value $y$ and is supposed to come up with a $x$ such that $H(x) = y$.)

3. Is $H$ second preimage resistant? If yes, explain why. If not, show an attack against the scheme. (The adversary is given a value $x'$ and is supposed to come up with a $x$ such that $x \neq x'$ and $H(x) = H(x')$.)

**Solution.** The answer is no in all three cases. Because an attack on second preimage resistance is also an attack on collision resistance, we will only give attacks on preimage resistance and second preimage resistance.

For preimage resistance, we are given a value $y$ and are supposed to find an $x$ so that $H(x) = y$. The outputs of $H$ are 128 bits long by design. We know the key $k$ of the block-cipher $E$. Thus we are able to decrypt any message. Let’s compute $z = D_k(y)$. $z$ is 128 bits long. Now $H(z) = E_k(z) = y$. Thus $z$ is a preimage of $y$.

As for second-preimage resistance, suppose we are given some $x' = x_1||x_2|| \ldots ||x_t$. The hash of it is $E_k(x_1) \oplus E_k(x_2) \oplus \ldots \oplus E_k(x_t)$. Because we are able to decrypt, let’s compute $z' = D_k(00 \ldots 0)$. Now let’s take $x' = x||z'$. We have that $128$ zeroes.

$$H(x') = E_k(x_1) \oplus E_k(x_2) \oplus \ldots \oplus E_k(x_t) \oplus E_k(z') =$$

$$= E_k(x_1) \oplus E_k(x_2) \oplus \ldots \oplus E_k(x_t) \oplus 00 \ldots 0 = E_k(x_1) \oplus E_k(x_2) \oplus \ldots \oplus E_k(x_t) = H(x).$$

Thus we have found a second preimage.

Alternatively, for second-preimage resistance, suppose that we are given some $x' = x_1||x_2$ with $x_1 \neq x_2$. We take $x = x_2||x_1$. The hash of both $x$ and $x'$ is $E_k(x_1) \oplus E_k(x_2)$. 

---

2