Homework 5

Due date: Thursday, November 30th 9:00 AM

Question 1 (25 points)

Consider the following protocol for two parties A and B to flip a fair coin.

1. A trusted party T publishes her public key pk;
2. Then A chooses a uniform bit b_A, encrypts it using pk, an announces the ciphertext c_A to B and T;
3. Next, B acts symmetrically and announces a ciphertext $c_B \neq c_A$;
4. T decrypts both c_A and c_B, and the parties XOR the results to obtain the value of the coin.

- Argue that even if A is dishonest (but B is honest), the final value of the coin is uniformly distributed.
- Assume the parties use EI Gamal encryption (where the bit b is encoded as the group element g^b before being encrypted — note that efficient decryption is still possible). Show how a dishonest B can bias the coin to any values he likes.
- Suggest what type of encryption scheme would be appropriate to use here. Can you define an appropriate notion of security for a fair coin flipping and prove that the above coin flipping protocol achieves this definition when using an appropriate encryption scheme?

Question 2 (15 points)

Secret sharing is a problem in cryptography where n shares $X_1, ..., X_n$ (called shadows) are given to n parties where some of the shadows or all of them are needed in order to reconstruct the secret (M) which is a number (i.e. there is a specified threshold t, such that any t shadows make it possible to compute M which is a bit string). Consider the following secret sharing algorithm:

1. Choose at random $t-1$ positive integers $a_1, ..., a_{t-1}$ with $a_i < P$ (P is a prime number) and let $a_0 = M$.
2. Build the polynomial $f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + ... + a_{t-1}x^{t-1}$.
3. Create n shadows that are: $(1, f(1)(\mod P)), ..., (n, f(n)(\mod P))$ (i.e. every participant is given a point (an integer input to the polynomial, and the corresponding integer output).

Note: Suppose $t < P - 1$
Based on the above protocol, answer the following questions:

(a) In above protocol, arithmetic is all modulo p to build the polynomial. Suppose that we mistakenly calculate the shadows as $(x, f(x))$ instead of $(x, f(x)(\mod P))$, can an eavesdropper gain information from M or not if the eavesdropper sees some of the points (e.g. Suppose the eavesdropper finds $(1, f(1))$ or $(2, f(2))$)? If your answer is no, please prove it otherwise provide an example that shows the eavesdropper can gain information about M.

(b) Suppose we modify the scheme such that $M = a_0 + a_1 + \ldots + a_{t-1} \mod p$. Does having t or more shadows make it possible to compute M ? Does having fewer than t shadows reveal nothing about M? Please justify your answers.

Question 3 (15 points)

Consider a variant of DSA in which the message space is $\mathbb{Z}_q$ and H is ommitted (i.e. the second component of the signature now $s := [k^{-1} \cdot (m + xr) \mod q]$). Show that this variant is not secure.

Question 4 (30 points)

Let f be one-way permutation. Consider the following signature scheme for messages in the set $\{1, ..., n\}$:

- $\mathsf{Gen}(1^n)$: choose uniform $sk \in \{0,1\}^n$ and set $pk := f^{(n)}(sk)$ (Where $f^{(i)}(\cdot)$ refers to i-fold iteration of f, and $f^0(x) \stackrel{\text{def}}{=} x$)
- $\mathsf{Sign}(m, sk)$: to sign $m \in \{1, \ldots, n\}$, output $\sigma = f^{(n-m)}(sk)$
- $\mathsf{Ver}(m, \sigma, pk)$: verify $pk \stackrel{?}{=} f^{(m)}(\sigma)$

(a) Show that the above is not a one-time-secure signature scheme. Given a signature on a message i, for what messages j can an adversary output a forgery?

(b) Prove that no ppt adversary given a signature on i can output a forgery on any message $j > i$ except with negligible probability.

(c) Suggest how to modify the scheme so as to obtain a one-time-secure signature scheme.

Question 5 (15 points)

Suppose that Alice has a secret bit a and Bob has a secret bits b_1, b_2 and that Alice and Bob want to compute the function $h(a, b_1, b_2) = b_1 \wedge (b_2 \oplus a)$ using Yao's Garbled Circuit protocol.

(a) Suppose that Alice selects two random permutations $\pi_1, \pi_2 : \{(0,0), (0,1), (1,0), (1,1)\} \to \{(0,0), (0,1), (1,0), (1,1)\}$. Write down the garbled circuit that Alice sends Bob.

(B) Suppose that Alice is malicious, but Bob behaves honestly during the execution of the protocol. Write down a garbled circuit that Alice can send Bob to extract the secret bit b_1 directly.

Bonus (10 points)

Let $pk = (N, e)$ (resp. $sk = (N, d)$) denote the public (resp. private) key in a plain RSA signature scheme. Define the function $\mathbf{Int} : \{0,1\}^* \rightarrow \mathbb{Z}_N^*$ as follows: on input string $x = (x_1 \| ... \| x_n) \in \{0,1\}^t$ we set

$$\mathbf{Int}\,(x_1 \| ... \| x_n) = \sum_{i=1}^{n} 2^{n-i} x_i$$

We also let μ denote an ASCII character to byte mapping in which $\mu(0) = 0^8, \mu(1) = 0^7 1, \mu(2) = 0^6 10, \ldots, \mu(9) = 0^4 1001$. Given an ASCII message $m = m_1, \ldots, m_n$ we define $\mathbf{Encode}(m) = \mathbf{Int}\,(\mu(m_1) \| ... \| \mu(m_n))$.

Finally, for and ASCII message m we can set

$$\mathbf{Sign}_{sk}\,(m) = \mathbf{Encode}\,(m)^d \mod N \ .$$

$\mathbf{Verify}_{pk}\,(m, \sigma)$ returns 1 if and only if $\sigma^e = \mathbf{Encode}\,(m)$.

Suppose Alice signs the message $m =$ "Please pay Bob the following ammount from my bank account (USD): 50." Suppose that Bob obtains $\sigma = \mathbf{Sign}_{sk}\,(m)$. Explain how Bob can obtain a signature σ' authorizing the bank to transfer more than \$50. How much money can Bob make? Assume that the Bank denies transfers above 750 million (USD) without in person authorization. You may assume that $\mathbf{Encode}(m) < N/2^{64}$.