University of Maryland
CMSC456 — Introduction to Cryptography
Professor Jonathan Katz

Problem Set 5

Due at the *beginning* of class on Dec. 2

1. Let p be a prime with $p = 3 \bmod 4$. Let $x \in \mathbb{Z}_p^*$ be a quadratic residue. Show how to compute the square roots of x *efficiently*. (*Hint*: Show how to find an even integer i such that $x^i = x$. Then $x^{i/2}$ is one square root of $x\ldots$) Use your algorithm to compute the square roots of 2 modulo 71.

2. Let $N = pq$ where p and q are distinct, odd primes. Prove that $\mathcal{L}_p(xy) = \mathcal{L}_p(x) \cdot \mathcal{L}_p(y)$ for $x, y \in \mathbb{Z}_p^*$ (where $\mathcal{L}_p(x)$ is the *Legendre symbol* of x, defined to be $+1$ if x is a quadratic residue modulo p, and -1 otherwise). Prove that $\mathcal{J}_N(xy) = \mathcal{J}_N(x) \cdot \mathcal{J}_N(y)$ for $x, y \in \mathbb{Z}_N^*$.

3. Let p, q be distinct primes with $p = q = 3 \bmod 4$. Recall the encryption scheme given in class based on the quadratic residuosity assumption: the public key is $N = pq$ and to encrypt a "0" the sender sends a random quadratic residue, while to encrypt a "1" she sends a random non-quadratic residue with Jacobi symbol $+1$.

   Show that given a public key N and ciphertexts C_1, C_2 which are encryptions (both with respect to N) of bits b_1, b_2, it is possible to efficiently compute a ciphertext C' which is an encryption of $b_1 \oplus b_2$ *without knowing the values* b_1, b_2. Show that given a public key N and a ciphertext C which is an encryption of some bit b, it is possible to efficiently generate a second, random ciphertext C' which is also an encryption of the same bit b (again, without knowing the value of b).

4. Assume two students Alice and Bob want to find out whether they both like each other. We will model this by assuming that Alice has an input bit x_A which is 1 iff she likes Bob, and similarly for Bob; the students want to compute the value of $f(x_A, x_B) = (x_A \text{ and } x_B)$. Since neither one wants to risk embarrassment, they want to find out the answer without (necessarily) revealing their true feelings; in particular, if $f(x_A, x_B) = 0$ and $x_B = 0$ (i.e., Bob does not like Alice) then Bob should not learn whether or not $x_A = 1$ (i.e., whether or not Alice likes Bob); similarly for Alice.

   A third student Carol has offered to help. She generates two random, distinct primes p and q with $p = q = 3 \bmod 4$ and announces the value $N = pq$. Carol then agrees to decrypt any *one* ciphertext that Alice and Bob give her (using the encryption scheme described in the previous question). Let $\mathcal{E}_N(b)$ denote the encryption of a bit b. Consider the following protocol:

   (a) Alice computes $C_A \leftarrow \mathcal{E}_N(x_A)$ and sends C_A to Bob.

   (b) Bob does the following: if $x_B = 0$, he computes $C_B \leftarrow \mathcal{E}_N(0)$. If $x_B = 1$, he sets $C_B = C_A$. In either case, he then sends C_B to Carol.

(c) Carol decrypts C_B and announces the result.

(1) Show that the final result computed by the above protocol is correct. (2) Show that Bob learns nothing about the input of Alice when $x_B = 0$. (3) Show that Alice *does* potentially learn something about the input of Bob when $x_A = 0$ (note that Alice sees what Bob sends to Carol). (4) Suggest a way to modify the protocol so that it is still correct, but neither Alice nor Bob learn anything about the other's input (when their own input is 0).

5. One of the more interesting applications of cryptography is that it enables secure gambling over the Internet. As an example, say parties A and B, communicating over the Internet and not in the same city, want to flip a fair coin. A will pay B \$1 if the coin is "0", and B will pay A \$1 if the coin is "1". Consider the following ways they might do this:

   (a) A flips a coin and sends the result to B.

   (b) A videotapes herself flipping a coin, and sends the result to B along with the video so he can verify that the coin-flip occurred as claimed.

   (c) A generates random primes p, q with $p = q = 3 \bmod 4$, chooses a random bit b, and sends $(N, \mathcal{E}_N(b))$ to B (the notation is like in the previous problem). Then B chooses a random bit b' and sends b' to A. Finally, A reveals p and q (at which point B can figure out b) and the value of the coin is $b \oplus b'$.

   (d) A generates random primes p, q with $p = q = 3 \bmod 4$, chooses a random bit b, and sends $(N, \mathcal{E}_N(b))$ to B. Then B chooses a random bit b' and sends $\mathcal{E}_N(b')$ to A. Finally, A reveals p and q (at which point everyone can figure out b, b') and the value of the coin is $b \oplus b'$.

   In which of the above protocols can A cheat and bias the value of the coin? In which of the above protocols can B cheat? Which of the above protocols (if any) are secure against cheating by either player?

6. Consider the following way to deal cards to two players A and B. In what follows, let $p = 2q + 1$ where p, q are prime, and let g be a generator of the subgroup of quadratic residues in $\mathbb{Z}_p^*$ (assume the decisional Diffie-Hellman assumption holds in this subgroup). All computations in what follows are done modulo p. Assume A has public key $y_A = g^{x_A}$ and B has public key $y_B = g^{x_B}$ where $x_A, x_B \in \mathbb{Z}_q$.

   (a) The deck of cards is some fixed set $\{y_1, \ldots, y_{52}\}$, where $y_1, \ldots, y_{52} \in \mathbb{Z}_p^*$ are distinct quadratic residues known to both A and B.

   (b) A begins by encrypting all cards using her public key; i.e., for each i she computes $C_i = \langle g^{r_i},\ y_A^{r_i} \cdot y_i \rangle$ (where all r_i are chosen independently at random from $\mathbb{Z}_q$). She then randomly permutes the set of ciphertexts and sends them to B.

   (c) B chooses five of the $\{C_i\}$ at random, call these $C_1^*, \ldots, C_5^*$, and sends them back to A (these will represent A's cards). From the remaining values, B chooses another five at random; call these $\hat{C}_1, \ldots, \hat{C}_5$. For each of these values $\hat{C}_i = \langle z_{i,1}, z_{i,2} \rangle$, B computes $\hat{C}_i' = \langle g^{r_i'},\ z_{i,1} \cdot g^{s_i},\ z_{i,2} \cdot y_B^{r_i'} \cdot y_A^{s_i} \rangle$ and sends $\hat{C}_1', \ldots, \hat{C}_5'$ back to A as well. Here, the r_i', s_i are chosen independently at random from $\mathbb{Z}_q$.

(d) For each of the values $\hat{C}'_i = \langle w_1, w_2, w_3 \rangle$ that A receives, A computes $\hat{C}''_i = \langle w_1,\ w_3/w_2^{x_A} \rangle$ and sends the result back to B (these will represent B's cards).

Show how both A and B can compute the value of the cards in their hand. Show that the resulting protocol is *correct* in that — assuming both A and B are honest — the 10 cards dealt are all distinct and uniformly-chosen from the deck. Argue also that neither A nor B know about the cards in the other player's hand.

Assume we change the protocol so that B can choose the set of cards $\{y_1, \ldots, y_{52}\}$ any way he wants, as long as $y_1, \ldots, y_{52} \in \mathbb{Z}_p^*$ and they are all distinct. Show how B can "mark" cards so that he can cheat when selecting his hand later (e.g., he can "mark" the ace of spades and make sure he always gets it). *Hint:* El Gamal encryption is secure only if it is used to encrypt quadratic residues...