University of Maryland
CMSC456 — Introduction to Cryptography
Professor Jonathan Katz

Problem Set 5

Due at *beginning* of class on Nov. 27

1. Consider the multiplicative group $\mathbb{Z}_p^*$, for p prime (recall also that $|\mathbb{Z}_p^*| = p - 1$). We mentioned in class that this is always a cyclic group; so fix some generator g for $\mathbb{Z}_p^*$. Also, the discrete logarithm problem in $\mathbb{Z}_p^*$ is conjectured to be hard.

   (a) Prove that g cannot be a quadratic residue in $\mathbb{Z}_p^*$. (Hint: if g is a quadratic residue, show that it cannot possibly generate the entire group).

   (b) Prove that the set $\{g^0, g^2, g^4, g^6, \ldots, g^{p-3}\}$ corresponds to the set of quadratic residues in $\mathbb{Z}_p^*$.

   (c) Define $\mathsf{CDH}_g(h_1, h_2) = g^{(\log_g h_1)\cdot(\log_g h_2)}$ (note that this function is well-defined, even if it cannot be efficiently computed). Give necessary and sufficient conditions on h_1 and h_2 for $\mathsf{CDH}_g(h_1, h_2)$ to be a quadratic residue.

   (d) If h_1 is chosen uniformly at random from $\mathbb{Z}_p^*$, what is the probability that it is a quadratic residue? If h_1 and h_2 are chosen independently and uniformly at random from $\mathbb{Z}_p^*$, what is the probability that $\mathsf{CDH}_g(h_1, h_2)$ is a quadratic residue?

   (e) Give an explicit (efficient) algorithm which shows that the DDH assumption *does not hold* in $\mathbb{Z}_p^*$. Analyze the success of your algorithm in distinguishing Diffie-Hellman quadruples from random quadruples. (Hint: use parts (c) and (d) and the fact that there exists an efficient algorithm to determine whether an element in $\mathbb{Z}_p^*$ is a quadratic residue or not.)

2. Let $h_N : \mathbb{Z}_N^* \rightarrow \{0, 1\}$ be a hard-core bit for RSA (so that, given $x^3 \bmod N$ it is hard to predict $h(x)$ with probability better than $1/2$). We showed in class that the following encryption scheme is secure: the public key is N, the private key is d for which $3d = 1 \bmod \varphi(N)$, and encrypting a bit b is done by choosing a random r and sending $(r^3 \bmod N, h_N(r) \oplus b)$.

   Say I want to send messages b_1, b_2, b_3 to each of three users with public keys N_1, N_2, and N_3, where N_1, N_2, N_3 are all different.

   (a) Show that if there exist distinct $i, j \in \{1, 2, 3\}$ with $\gcd(N_i, N_j) \neq 1$ then an adversary can factor N_i and hence decrypt my message to user i.

   (b) Say I use the same r to encrypt messages for each user. So I send $(r^3 \bmod N_1, h_{N_1}(r) \oplus b_1)$, $(r^3 \bmod N_2, h_{N_2}(r) \oplus b_2)$, and $(r^3 \bmod N_3, h_{N_3}(r) \oplus b_3)$. Show that an adversary, given just $r^3 \bmod N_1$, $r^3 \bmod N_2$, and $r^3 \bmod N_3$, can efficiently recover r and hence decrypt my messages to all the users. (Hint: Use Chinese remaindering modulo $N_1 N_2 N_3$.)

3. Consider the following modification of the El Gamal encryption scheme over group G: the public key is (g, h), the secret key is $\log_g h$, and message $m \in \{0, \ldots, |G| - 1\}$ is encrypted by choosing random r and sending $(g^r, h^r g^m)$.

   (a) Show how the receiver can recover g^m.

   (b) If the discrete logarithm problem is hard in G, recovering g^m will not, in general, allow the receiver to recover m. Argue that if we assume the sender only sends messages $m \in \{0, \ldots, 100\}$ then the receiver *can* recover m. Will the scheme be secure if we restrict m in this way?

   (c) Say (A_1, B_1) is an encryption of m_1. Prove that $(A_1, B_1 \cdot g^{m_2})$ is an encryption of $(m_1 + m_2) \bmod |G|$.

   (d) Say (A_1, B_1) is an encryption of m_1 and (A_2, B_2) is an encryption of m_2. What is $(A_1 A_2, B_1 B_2)$ an encryption of?

   (e) Assume the receiver R is conducting an auction in which two bidders each encrypt their bids and send them to R. The bid of the first bidder is assumed to be in the range $\{0, \ldots, 100\}$. Argue that the bidder who goes second can cheat and always win by bidding \$1 more than the first bidder *even without ever learning the value of the first bidder's bid.*