

Problem Set 5

Submit this problem set in PostScript, PDF, or MS Word format to `6857-submit@csail.mit.edu` before lecture on the due date. We have provided templates for L^AT_EX and Microsoft Word on the course website. Each solution must appear on separate sheets of paper. Mark the top of each sheet with your name(s), the course number (6.857), the problem set number and question, and the date.

You are to work in groups of three or four people and should submit a single set of solutions for all problems parts designated **[Group]**. You should turn in a separate, individual solution to any problems designated **[Individual]**.

Problem 5-1. Secret Sharing [Group]

A master secret $s \in \mathbb{Z}_p$, where p is a large prime, was split into 22 shares by selecting a random polynomial $P(x)$ of degree 4, where $P(0) = s$. This polynomial is evaluated over the field \mathbb{Z}_p , that is, every operation is taken modulo p . We denote this as $P(x) \in \mathbb{Z}_p[x]$.

Once $P(x)$ was selected, 22 shares of s were generated by evaluating $P(x)$ at the points $[1, 22]$ to obtain $P(1), \dots, P(22)$. Since $P(x)$ has degree 4, any 5 different shares can fully reconstruct the polynomial P . In other words, s has been split using a (22, 5)-Shamir Secret Sharing scheme.

Each group will receive an individual share $P(i)$. However, this share will be split a second time among your t -member group by choosing a random polynomial $Q(x) \in \mathbb{Z}_p[x]$ of degree $d = \max(1, t - 2)$, such that $Q(0) = P(i)$. Each member will receive a share $Q(1), \dots, Q(t)$ via e-mail. As usual, $d + 1$ shares can reconstruct $P(i)$.

Once your group recovers $P(i)$, you'll need to find four other groups to recover s . To make this problem fun and interesting, it is against the rules for groups to help each other in any way except trading their own $P(i)$ shares. For example:

1. Do not tell another group any of your group's $Q(j)$ values.
2. Do not share your group's $P(i)$ value unless explicitly asked for it. You are not allowed to post it to the class mailing list or to a website.
3. Do not re-transmit another group's $P(i)$ values.
4. Do not notify another group that they incorrectly calculated their $P(i)$ share.
5. Do not tell another group the master secret s or any partial information about s .

Exactly one group may have been given an incorrect value for $P(i)$. Be careful! Yours may be this group! And other groups may have incorrect software. However, groups caught intentionally cheating by giving incorrect values will be penalized. You can acquire more than 5 $P(i)$ shares to verify your s value if necessary.

Every student should receive their group's number i , their own own $(j, Q(j))$ pair, and the global prime p via e-mail. Hint: The value s is a prime number and its decimal representation will contain the string "6857".

Turn in the following:

- Your group's $P(i)$ value.
- The $(j, P(j))$ values of other group shares you used to compute s .
- The master secret s .
- If you think you received a fake $P(i)$ value, a brief explanation of how you know this is the case.
- A sentence or two attesting that your group did not violate the given rules or aid another group in computing s .

Problem 5-2. RSA and Semantic Security [Individual]

Throughout this problem, you may assume that RSA is computationally hard. That is, assume that no probabilistic polynomial time algorithm can decrypt an RSA ciphertext without knowledge of the private key with more than negligible probability.

- (a) In one sentence, explain why the original RSA is not semantically secure.
- (b) Implementations of RSA often choose 3, 5, 17, or 65537 as their exponent e . Assume that for each of these e values, $e \in \mathbb{Z}_{\phi(n)}^*$. In two sentences, explain why these particular values make better RSA exponents than a random element $e \in \mathbb{Z}_{\phi(n)}^*$.
- (c) For convenience, assume our RSA exponent e is 3 and that $e \in \mathbb{Z}_{\phi(n)}^*$. A proposal to make RSA semantically secure is as follows: Let $|x|$ signify the number of bits representing a number x . If $k = |n|$, define the length of a valid message m to be $|m| < 3k/4$. For each encryption operation, choose $k/4$ random bits r , append them to m such that $m' = (m \circ r) = m * 2^{k/4} + r$ (where \circ is the composition operator), and encrypt the value m' . Note that $|m'| < |n|$, so $m' \in \mathbb{Z}_n$. When we decrypt a ciphertext, we will simply discard the $k/4$ least significant bits of the decrypted plaintext. Is this scheme semantically secure? Prove or disprove it either way.
- (d) Suppose we have a standard RSA public/private keypair. Consider an RSA-variant where we will first select a random $r \in \mathbb{Z}_n^*$. To encrypt a message $m \in \mathbb{Z}_n^*$, we will compute a ciphertext of the form $c = (m \oplus r, r^e \bmod n) = (s, t)$, where \oplus is the binary XOR operator. To decrypt c , one can simply decrypt t using RSA and XOR the result with s . Is this scheme semantically secure? Prove or disprove it either way.

Problem 5-3. Notorious BIGnum [Group]

Professor Bignum is taking a sabbatical to write a book on security, and doesn't want to be disturbed much. So he devises the following scheme.

He decides that the only folks who should be able to send him email are his mother Alice, his wife Bobbie, or his daughter Charlene. But he wants to require that at least two of them must cooperate to send him email.

He plans to create an RSA public-key instance for himself, with public exponent e , modulus n , and secret key d . He will register and publish his public key with the certificate authority VeriKey.

He next plans to create six values e_1, e_2, e_3, e_4, e_5 , and e_6 , where e_1, e_3 , and e_5 will be chosen randomly, and e_2, e_4 , and e_6 will be found satisfying the equations:

$$\begin{aligned} e_1 e_2 &= e \bmod \phi(n) \\ e_3 e_4 &= e \bmod \phi(n) \\ e_5 e_6 &= e \bmod \phi(n) \end{aligned}$$

He then will give Alice e_1 and e_3 , gives Bobbie e_2 and e_5 , and gives Charlene e_4 and e_6 . Thus, any two of them can encrypt a message for Professor Bignum. For example,

$$m^e = (m^{e_1})^{e_2} \bmod n$$

when Alice and Bobbie cooperate. Alice and Charlene use e_3 and e_4 , and Bobbie and Charlene use e_5 and e_6 . Professor Bignum then starts to write his email handler so that it rejects any email that does not appear to be encrypted with RSA using exponent e .

Explain as many things as you can that are wrong with Professor Bignum's plan.

Problem 5-4. Block Cipher Modes [Group]

A message is sent using a block cipher without any authentication mechanism. An adversary can see the ciphertext (along with any initializing information that might be required) and modify it as it is transmitted. You are to evaluate the electronic code-book (ECB), cipher block-chaining (CBC), and counter (CTR) modes of operation of the cipher.

Consider an adversary who wishes to introduce a few controlled, single bit errors into the messages. That is, he may wish to flip the fourteenth bit in the third block of the message, not necessarily knowing what that message is. (Of course, the adversary doesn't know the encryption key.)

- (a) Give one realistic example of when such an attack would be desirable.
- (b) Under which modes of operation can arbitrary changes of this kind be made within a block? How is this performed? What side effects, if any, would this have?

Now assume that the adversary wishes to work with blocks as a whole. That is, although he could just send a block he created, that block would likely decrypt to gibberish. Forgoing the targeted bit errors of the previous section, he wants to use the blocks which he knows form a meaningful plaintext in order to construct other possibly meaningful plaintexts.

- (c) If each of the modes (ECB, CBC, CTR) is used for the original message, what new, valid messages can the adversary now construct? How?
- (d) Consider a combination of encryption and authentication that uses CBC mode encryption on the message, then runs a CBC-MAC on the ciphertext to get a MAC. The same key and IV are used as inputs to both the encryption and the MAC. Assess the security of this scheme.