National University of Singapore

CS4236 Cryptography Theory and Practice
(Semester 1: AY2018/19)

Time Allowed: 2 Hours

INSTRUCTIONS TO STUDENTS:

- Please write your Student Number only. Do not write your name.
- This assessment paper contains NINE questions and comprises FIFTEEN printed pages.
- Students are required to answer ALL questions.
- Students should write the answers only on the front pages in the designated papers (right below the questions and often the next page as well). Anything on the backside of the papers will be ignored.
- Only 1 A4-size help sheet is allowed.
- Any collaboration or assistance of any kind (including phones, tablets, calculators, or laptops) is strictly prohibited; all work must be your own.
- Hint: in general, shorter answers are better than longer answers. Spend more time thinking and less time writing.

STUDENT NUMBER: _____

(Write your Student Number in *all* the 15 pages)

Questions	Graded Points	Remarks
Q1	/15	
Q2	/12	
Q3	/12	
Q4	/23	
Q5	/18	
Q6	/5	
Q7	/7	
Q8	/7	
Q9	/1	
Total	/100	

1 CBC Encryption and CBC MAC (15 points)

Suppose the same symmetric key is used for message encryption in the standard CBC mode and for arbitrary-length CBC-MAC computations. You are an adversary who does not know the symmetric key used for either CBC encryption or CBC-MAC computations.

a) **(7 points)** Can an attacker forge messages protected by standard CBC-MACs (with IV= 0) from chosen plaintext-ciphertext pairs obtained by encryption in the standard CBC mode (using a random IV output as ciphertext block c_0)? If yes, show an attack; otherwise, explain why. If the sender chooses arbitrary IV for CBC-MAC and sends it along with the MAC tag, how would your answer be changed?

b) **(8 points)** Is it possible to forge a CBC encryption (i.e., a ciphertext of a CBC encrypted message) via a chosen-plaintext attack using tags of CBC-MACs? If yes, show an attack. If no, explain why.

Answer for Q1:

CBC is not deterministic cos of IV
CBC-MAC is deterministic, MACs should never be randomised too as its purpose is to verify

CBC encryption: ciphertext is formed by blocks of c1c2...cn
CBC-MAC: only the last output, which is a hash, is used as a MAC

CBC-MAC is actually a pseudorandom function for message of size l(n) => MAC-forge secure
sender and receiver agree on size l(n) block
If not l(n) => not MAC-forge secure
Arbitrary but fixed-length => can

(a)
Basically this: https://crypto.stackexchange.com/questions/51866/having-cbc-mac-of-a-one-block-message-forge-cbc-mac-of-two-blocks
Arbitrary IV for CBC-MAC will defeat the purpose of MAC => non-deterministic => cannot verify

2 RSA Encryption (12 points)

a) **(5 points)** Given an RSA public-key (N,e), show that the problem of finding d such that $ed = 1 \bmod \phi(N)$ equivalent to the problem of factoring N (i.e., both imply each other).

b) **(7 points)** Alice generates a public RSA key (N, e) (and related private key d). Being overly paranoid, she shares the private keys d_b and d_c (such that $d = d_b \times d_c \bmod \phi(N)$) with her best friends Bob and Carol, in case she is captured by aliens.

(1) Show how Bob and Carol together can decrypt a ciphertext c intended to Alice without revealing d_b and d_c to each other. **Mulit-Party Computation**

(2) Also, show how Bob and Carol can convince each other that they jointly have the power to decrypt ciphertexts intended for Alice without revealing d_b or d_c to each other, before such a ciphertext is given to them.
Verifiable Secret Sharing (VSS)

Answer for Q2:

a) If she can find d based on the given RSA public key, she can factor N. Lec 10 Slide 38, 39

Algo that factors N only with N and e, runs at the same time as an algo that finds d.

If you factor n you can easily obtain d from e, and if you somehow obtained d and e you can easily factor n.

(Show steps using examples)

https://crypto.stackexchange.com/questions/65987/why-cant-we-find-d-if-we-know-e-and-n-in-rsa

b) Use Multi-Party Computation

Both Bob and Carol act as dealers to one another
Bob sends a share of his db to Carol via secret sharing
Carol sends a share of her dc to Bob via secret sharing

Bob and Carol both pool their shares together and reconstruct.

c) Bob must prove to Carol that he is not a dishonest dealer. Bob sends Carol his share along with extra elements, allowing Carol to verify his share (constructed from db) from the extra element.
The same is done with Carol to Bob.

If shares reconstruct to d => can decrypt ciphertext => no need for ciphertext.

db dc not revealed cos of using shares,

3 Diffie-Hellman Key Exchange (12 points)

a) **(3 points)** Alice picks a 2048-bit prime p and uses $\mathbb{Z}_p^*$ for the group for Diffie-Hellman key exchange. Describe the security problem she would face and explain how can she fix. We assume an authenticated channel between two parties.

b) **(4 points)** Show why the Decision Diffie-Hellman assumption implies the Discrete Logarithm assumption and not the other way around.

c) **(5 points)** Four people, Alice, Bob, Carol, and David, want to agree on a common key. They publicly choose a large prime p and a generator g. They privately choose random numbers a, b, c, and d, respectively. Describe a protocol that allows them to compute $k = g^{abcd} \bmod p$ securely (ignoring man-in-the-middle attacks). MPC and secret sharing

Answer for Q3:

a)
Can use Legendre symbol to identify g^ab mod p, given g^a mod p, g^b mod p.
Use Zn, and use g^2 generator.

b) DDH: Distinguish g^ab from g^z given <g^a, g^b>
DDH-Hard => DL assumption holds
~ DL assumption holds => ~ DDH-Hard
If Adv obtains a by taking the DL of g^a, and obtains b by taking the DL of g^b => Adv can efficiently calcate g^ab, differentiating g^ab from g^z, which is not DDH-hard anymore.

DL assumption holds => DDH-Hard (this is a converse error based on the above)
Hence,
DL assumption holds ≠> DDH-Hard

c)
MPC because they each privately chose a random num
g^a * g^b = g^ab
But this causes a dimension increase
Each of the 4 must perform subsidiary sharing, where they compute half the dimension for the next j.

t = 4 (because 4 values => 4 points on the curve)
Let h(x) be a polynomial of degree 4t - 4 = 12
Let r(x) be a polynomial of degree 3 (so that 4 points represented by 4 values determine 1 curve, which when intersected with the y-axis, gives k)

Each of the 4 shares with each other a share of g^i where i is from the set {g^a, g^b, g^c, g^d mod p} .
Each of the 4 computes h(j) = (s1 x s2 x s3 x s4) mod p, and distributes shares of h(j) to everyone. j is the number corresponding to the person.
Each of the 4, after receiving the shares of h(j), computes its share for r(j) for all j, where j is from the set of shares from h(j).
r(x) can then be plotted by each person and the intersection at the y-axis yields k = g^abcd mod p

4 Hash and Digital Signatures (23 points)

a) **(5 points)** Alice constructs her own hash function h that takes a fixed-length 120-bit $x = x_1 \| x_2$ (56-bit x_1 and 64-bit x_2) and outputs a 64-bit y. Instead of building it from scratch, she decides to use DES block cipher $y = \mathsf{DES}_{x_1}(x_2)$ (Recall that DES takes a 56-bit key and a 64-bit input block, and outputs a 64-bit output block). Argue:

- whether h is secure against preimage attacks;
- whether h is secure against second preimage attacks; and
- whether h is secure against collision attacks.

Explain your answers.

b) **(8 points)** Bob also tries to construct his own hash function using RSA encryption. Consider using RSA with a known key. Let say a message consists of a sequence of blocks $B_1, B_2, \cdots, B_n$. Then, Bob encrypts the first block, XOR the result with the second block and encrypt again. Bob repeats this for all the message blocks. That is, $H(B_1 \| B_2 \| \cdots \| B_n) = RSA(\cdots RSA(RSA(B_1) \oplus B_2)) \cdots B_n)$. Show that this does not satisfy second preimage resistance.

c) **(10 points)** Suppose Alice signed two documents m_1 and m_2 using Digital Signature Algorithm. She mistakenly used the same random per-message value k and generated the signature (r_1, s_1) and (r_2, s_2). Show how Bob, who doesn't knows the value of k, can figure out that Alice uses the same k. Also, show that Bob can derive the private key x.

Recall: DSA has public key $(p, q, g, y = g^x \bmod p)$ and private key $x \in \mathbb{Z}_q^*$.

- Sign: (1) choose $k \in \mathbb{Z}_q^*$ uniformly at random; (2) compute $r = (g^k \bmod p) \bmod q$; (3) compute $s = (h(m) + x \cdot r) \cdot k^{-1} \bmod q$ (h is a collision resistant hash function: $\{0,1\}^* \to \mathbb{Z}_q$)); (4) the signature for m is (r, s).
- Vrfy: (1) compute $u_1 = h(m) \cdot s^{-1} \bmod q$; (2) compute $u_2 = r \cdot s^{-1} \bmod q$; (3) output "yes" if and only if $r = (g^{u_1} y^{u_2} \bmod p) \bmod q$

Answer for Q4 (next page as well):

Answer for Q4 (cont'd):

5 ElGama Encryption (18 points)

Consider the ElGamal encryption scheme, where the plaintexts and ciphertexts are from the domain $\mathbb{Z}_p^* = \{1, \ldots, p-1\}$ and p is a prime. Recall that the public key is g, $h = g^x$, p and the private key is x where g is a generator. Given a plaintext m, the ciphertext is $E(m) = \langle mh^r, g^r \rangle$, where r is chosen uniformly from $\mathbb{Z}_p^*$ at random. All arithmetic operations are performed under modulo p.

a) **(3 points)** Note that 0 is not in the domain of plaintexts. Bob wants to include 0 as a plaintext and to apply the same encryption steps described above on 0. Explain why this is not desired.

b) **(5 points)** Given two tuples $t_1 = \langle c_1, b_1 \rangle$ and $t_2 = \langle c_2, b_2 \rangle$, let us define $t_1 t_2 = \langle c_1 c_2, b_1 b_2 \rangle$. Show that ElGamal can be homomorphic with respect to multiplication.

c) **(5 points)** Suppose Alice is able to solve CDH (Computational Diffie-Hellman) in $\mathbb{Z}_p^*$ that is, given g^a, g^b in $\mathbb{Z}_p^*$, she can find g_{ab}. Show that given the public key g, $h = g^x$, p and a ciphertext $\langle mh^r, g^r \rangle$ of ElGamal, Alice can find the plaintext.

d) **(5 points)** Which one of the following s is the most appropirate conclusion we can derive from question c)? For your choice of answer, describe the meaning of the term "secure".

i. ElGamal is "secure", assuming that CDH is difficult.

ii. In order for ElGamal to be "secure", CDH must be difficult.

iii. ElGamal is "secure" if and only if CDH is difficult.

ii).
secure means cannot find PT without private key with more than negl probability.

Answer for Q5 (next page as well):

Answer for Q5 (cont'd):

6 Transport Layer Security (5 points)

Consider the following threats to TLS security and describe how each is countered by a particular feature of TLS.

a) Brute-Force Cryptanalytic Attack: An exhaustive search of the key space for a conventional encryption algorithm.

b) Known Plaintext Dictionary Attack: Many messages will contain predictable plaintext, such as the `HTTP GET` command. An attacker constructs a dictionary containing every possible encryption of the known-plaintext message. When an encrypted message is intercepted, the attacker takes the portion containing the encrypted known plaintext and looks up the ciphertext in the dictionary. The ciphertext should match against an entry that was encrypted with the same secret key. If there are several matches, each of these can be tried against the full ciphertext to determine the right one.

c) Replay Attack: Earlier TLS handshake messages are replayed.

d) Man-in-the-Middle Attack: An attacker interposes during key exchange, acting as the client to the server and as the server to the client.

e) Password Sniffing: Passwords in HTTP or other application traffic are eavesdropped.

Answer for Q6:

7 Homomorphic Encryption (7 points)

The Paillier encryption scheme:

- Gen: Generates (N, p, q). N is the public key, and $\langle N, \phi(N) \rangle$ is the private key.
- Enc: For a message $m \in \mathbb{Z}_N$, a random $r \leftarrow \mathbb{Z}_N^*$, output the ciphertext $c = (1+N)^m \cdot r^N \bmod N^2$.
- Dec: Compute $m = \frac{[c^{\phi(N)} \bmod N^2]-1}{N} \cdot \phi(N)^{-1} \bmod N$.

a) **(3 points)** Refer to the above encryption scheme and show that the Paillier encryption scheme is homomorphic. **ab = Dec(Enc(a) + Enc(b))**
ab = Dec(Enc(a) * Enc(b))

b) **(4 points)** SoC decides to use the Paillier encryption to vote for the school's new mascot. Each vote v_i can be either 0 (i.e., "no") or 1 (i.e., "yes"). A trusted authority (i.e., Dean of SoC) publishes a public key N for the Paillier encryption. The voter i casts her vote v_i by computing $c_i = [(1+N)^{v_i} \cdot r_i^N] \bmod N^2$ for a random $r_i \in \mathbb{Z}_N^*$. All the SoC students collectively calculate the final aggregated vote. The trusted authority receives the final aggregated vote and reveals the final vote total. Show how the votes are aggregated (assuming our students are honest and follow the protocol) and the trusted authority can compute the vote total without learning individual votes.

Answer for Q7:

8 Secret Sharing (7 points)

A dealer wants to share a secret $a_0 \in \mathbb{Z}_p$, in a 3-out-of-5 manner, with information-theoretical security. The dealer determines random coefficients a_1, a_2, defining the polynomial $q(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2$ and then starts distributing shares, sending $q(1)$ to party P_1 and $q(2)$ to party P_2. At this point, the dealer realizes that it wants instead to change the secret to $b_0 \in \mathbb{Z}_p$, with $b_0 \neq a_0$, but it can no longer change the choice of shares sent to P_1 and P_2. Is it still possible to choose a different polynomial q, such that if shares $q(3)$, $q(4)$, $q(5)$ are sent to parties P_3, P_4, P_5, respectively, the 5 overall shares constitute a 3-out-of-5 secret sharing scheme for secret b_0? If no, prove it. If yes, show the formula for $q(3)$, $q(4)$ and $q(5)$, based on $q(1)$, $q(2)$ and b_0.

Answer for Q8:

9 Final Checks (1 points) — LAST QUESTION

Have you double checked if you have your Student Number in all the 15 pages (including the scratch space)?

Answer for Q9: Yes or No.

Scratch space:

Scratch space:

Scratch space:

Scratch space (LAST SCRATCH PAGE):