CPSC 418 / MATH 318 — Introduction to Cryptography
ASSIGNMENT 3 — SOLUTION KEY

Written Problems for CPSC 418 and MATH 318

Problem 1 — A modified variant of the Diffie-Hellman protocol (6 marks)

Consider the following modification of the Diffie-Hellman protocol.

Set-up:

- All participants agree on a public prime p and a primitive root of p.
- Each participant generates their own random exponent u_i with $1 < u_i < p-1$ and computes $U_i \equiv g^{u_i} \pmod{p}$.

The public parameters g, p as well as the numbers U_i are stored in a public database. Each participant keeps their exponent u_i secret.

Suppose Alice and Bob wish to establish on a shared cryptographic key. Denote Alice's database entry by $X \equiv g^x \pmod{p}$ and Bob's database entry by $Y \equiv g^y \pmod{p}$. They proceed as follows:

Key Agreement Protocol

1) Alice generates a secret exponent a with $1 < a < p-1$, computes $A \equiv g^a \pmod{p}$ and sends A to Bob.

   Bob generates a secret exponent b with $1 < b < p-1$, computes $B \equiv g^b \pmod{p}$ and sends B to Alice.
2) Alice looks up Bob's public database entry Y and computes $K \equiv B^x Y^a \pmod{p}$.

   Bob looks up Alice's public database entry X and computes $K \equiv A^y X^b \pmod{p}$.

The key shared between Alice and Bob is K. The man-in-the-middle attack on ordinary Diffie-Hellman discussed in class no longer works here as long as the database is tamper-proof (of course any attacker with the ability to hack into the database can simply replace Bob's database entry by her own and impersonate Bob).

a. (2 marks) Formally prove that Alice and Bob compute the same quantity K in step 2.

   Solution.

$$B^x Y^a \equiv (g^b)^x (g^y)^a \equiv g^{bx+ya} \equiv g^{ay+xb} \equiv (g^a)^y (g^x)^b \equiv A^y X^b \pmod{p} \ .$$

   The left most expression is the result of Alice's computation and the right most expression is the result of Bob's computation. So both computations produce the same result.

b. (4 marks) Recall the Diffie-Hellman problem whose solution breaks the original Diffie-Hellman protocol:

   Given p, g, $A \equiv g^a \pmod{p}$ and $B \equiv g^b \pmod{p}$, compute $g^{ab} \pmod{p}$.

Suppose Eve is able to solve the Diffie-Hellman problem efficiently for any inputs p, g, A and B. Explain how she can use this capability when eavesdropping on Alice and Bob to find any key generated by Alice and Bob using the modified Diffie-Hellman protocol described above.[1]

Solution. Eve knows p, g and intercepts A, B via eavesdropping on Alice and Bob. She can also look up Alice's ad Bob's respective public database entries X, Y. She solves the Diffie-Hellman problem for inputs $A \equiv g^a \pmod{p}$ and $Y \equiv g^y \pmod{p}$ to obtain $g^{ay} \pmod{p}$. Similarly, she solves the Diffie-Hellman problem for inputs $X \equiv g^x \pmod{p}$ and $B \equiv g^b \pmod{p}$ to obtain $g^{xb} \pmod{p}$. Finally, she multiplies the results of her two Diffie-Hellman problem solution together to obtain

$$g^{ay} g^{xb} \equiv g^{ay+xb} \equiv K \pmod{p} \ .$$

[1] You may *not* assume here that Eve has the capability of extracting discrete logarithms; this is a stronger assumption. Remember that an algorithm for solving the discrete logarithm problem solves the Diffie-Hellman problem, and it is easy to see that it also makes finding keys for this variant very simple. But the reverse direction is unknown; that is, it is unknown whether anyone with the capability of discovering Diffie-Hellman keys or keys for this modified variant can use this ability to compute discrete logarithms.

Problem 2 — Properties of cryptographic hash functions (12 marks)

Recall the two properties of collision resistance and pre-image resistance that a cryptographic hash function needs to satisfy. In class, we asserted that these properties do not imply each other. This problem introduces two hash functions that confirm this fact; one is pre-image resistant but not collision resistant, the other is collision resistant but not pre-image resistant.

a. Let $h : \{0,1\}^* \to \{0,1\}^n$ be a pre-image resistant hash function. Define a new hash function $H : \{0,1\}^* \to \{0,1\}^n$ via

$$H(M) = h(M') ,$$

where $M' \in \{0,1\}^*$ is the bit string obtained by replacing the last bit of M by 0. That is, if the last bit of M is 0, then $M' = M$, whereas if the last bit of M is 1, then M' agrees with M except for the last but which is changed from 1 in M to 0 in M'.

(i) (4 marks) Formally prove that H is pre-image resistant.

Hint: Proof by contradiction works nicely here. Specifically, prove that if you can find a pre-image of some $x \in \{0,1\}^n$ under H, then you can also find a pre-image of $x \in \{0,1\}^n$ under h, contradicting the pre-image resistance of h. Also, don't overthink this; this is a really easy problem.

Solution. Assume to the contrary that H is not pre-image resistant. Then there exists $x \in \{0,1\}^n$ for which it is feasible to find a pre-image under H, i.e. a bit string $M \in \{0,1\}^*$ such that $H(M) = x$. Let M' be obtained from M by setting the last bit of M to 0. Then $h(M') = H(M) = x$. So we have efficiently found a pre-image M' of x under h, contradicting the pre-image resistance of h. Hence H is pre-image resistant.

(ii) (2 marks) Prove that H is not collision resistant by exhibiting an explicit example of a collision for H. Explain your example.

Solution. Put $M = 1$ and $M' = 0$, for example. Then $M \neq M'$. Replacing the last (in fact, the only) bit in M by 0 yields M', so $H(M) = H(1) = h(0) = H(0) = H(M')$. Hence $\{M, M'\}$ is a strong collision for H. (In fact, we have $H(S\|0) = H(S\|1)$ for all bit strings $S \in \{0,1\}^*$.

b. Let $h : \{0,1\}^* \to \{0,1\}^n$ be a collision resistant hash function. Define a new hash function $H : \{0,1\}^* \to \{0,1\}^{n+1}$ via

$$H(M) = \begin{cases} 1\|0^n & \text{if } M = 0, \\ 0\|h(M) & \text{otherwise,} \end{cases} ,$$

where, as usual, "$\|$" denotes string concatenation and $1\|0^n$ is the string of length $n+1$ that begins with a 1, followed by n 0's.

(i) (4 marks) Formally prove that H is collision resistant. *Hint:* Go with proof by contradiction again. Specifically, prove that if you can find a collision for H, then you can also a collision for h, contradicting the collision resistance of h.

Solution. Assume to the contrary that H is not collision resistant. Then it is feasible to find a two input strings $M, M' \in \{0,1\}^*$ with $M \neq M'$ and $H(M) = H(M')$.

Suppose first $H(M)$ begins with 1. Then $H(M')$ also begins with 1, as $H(M) = H(M')$. But the only string whose image under H begins with 1 is the string $1\|0^n$ which has the unique pre-image 0. This forces $M = 0$ and $M' = 0$, contradicting the fact that $M \neq M'$.

Hence $H(M)$ and $H(M')$ both begin with 0. It follows that $H(M) = 0 \| h(M)$ and $H(M') = 0 \| h(M')$, so

$$0 \| h(M) = H(M) = H(M') = 0 \| h(M') \ ,$$

which implies $h(M) = h(M')$. So we have found a collision for h, contradicting the collision resistance of h. Thus, H is collision resistant.

(ii) (2 marks) Prove that H is not pre-image resistant by exhibiting an explicit example of a string $x \in \{0,1\}^{n+1}$ for which it is easy to find a pre-image under H. Explain your example.

Solution. The string $x = 1 \| 0^n$ is an example of a hash value of H for which it is feasible to find a pre-image M, namely $M = 0$.

Problem 3 — CFB-MAC (6 marks)

Consider a block cipher whose plaintexts and ciphertexts are bit strings of length $n \in \mathbb{N}$, where encryption using a key K is denoted by E_K. Recall the the message authentication code CBC-MAC, where the MAC of a message M is the last ciphertext block when encrypting M using CBC mode:

CBC-MAC

1) Pad M with zeros so the length of the padded string is a multiple of the block length n.
2) Split the padded string into L blocks $M_1, M_2, \ldots, M_L$ where each block M_i has length n.
3) Put $C_0 = 0^n$ (the string of n zeros).
4) For $1 \leq i \leq L$, compute $C_i = E_K(M_i \oplus C_{i-1})$.
5) CBC-MAC$(M) = C_L$.

Similarly, define a message authentication code CFB-MAC based on using the same block cipher in CFB mode:[2]

CFB-MAC

1) Pad M with zeros so the length of the padded string is a multiple of the block length n.
2) Split the padded string into L blocks $M_1, M_2, \ldots, M_L$ where each block M_i has length n.
3) Put $C'_0 = M_1$ (the first message block)
4) For $1 \leq i \leq L-1$, compute $C'_i = M_{i+1} \oplus E_K(C'_{i-1})$.
5) CFB-MAC$(M) = E_K(C'_{L-1})$.

Now Let M be any message, K a key, C_i $(1 \leq i \leq L)$ the sequence of ciphertext blocks obtained when computing CBC-MAC(M) and C'_i $(1 \leq i \leq L-1)$ the sequence of ciphertext blocks obtained when computing CFB-MAC(M).

a. (4 marks) Use induction on i to prove that $C_i = C'_i \oplus M_{i+1}$ for $0 \leq i \leq L-1$.

Solution. For $i = 0$, we have $C_0 = 0^n$ and $C'_0 \oplus M_1 = M_1 \oplus M_1 = 0^n$, so $C_0 = C'_0 \oplus M_1$ as asserted.

Now suppose that $C_{i-1} = C'_{i-1} \oplus M_i$ for some i with $1 \leq i \leq L-2$. We need to prove that $C_i = C'_i \oplus M_{i+1}$. We have $C_{i-1} \oplus M_i = (C'_{i-1} \oplus M_i) \oplus M_i = C'_{i-1}$ and hence

$$C_i = E_K(M_i \oplus C_{i-1}) = E_K(C'_{i-1}) \ ,$$
$$C'_i \oplus M_{i+1} = \big(M_{i+1} \oplus E_K(C'_{i-1})\big) \oplus M_{i+1} = E_K(C'_{i-1}) \ ,$$

so $C_i = C'_i \oplus M_{i+1}$ as asserted. Part (a) now follows by induction.

b. (2 marks) Prove that CBC-MAC(M) =CFB-MAC(M).

Solution.

$$\text{CBC-MAC}(M) = C_L = E_K(M_L \oplus C_{L-1}) = E_K(C'_{L-1}) = \text{CBC-MAC}(M) \ .$$

[2] CFB-MAC can be defined more generally with any intitial value C'_0, in which case it differs from CBC-MAC. But for this problem, we specifically pick $C'_0 = M_1$.

Problem 4 — A variant of RSA (10 marks)

Let a, b be integers that are not both zero. The *least common multiple* of a, b, denoted $\operatorname{lcm}(a, b)$, is the unique positive integer satisfying the following properties:

(A) a divides $\operatorname{lcm}(a, b)$ and b divides $\operatorname{lcm}(a, b)$.

(B) If L is an integer such that a divides L and b divides L, then $\operatorname{lcm}(a, b)$ divides L.

You may use without proof the fact that the integer $\operatorname{lcm}(a, b)$ is unique for any a, b and satisfies

$$\gcd(a, b) \cdot \operatorname{lcm}(a, b) = ab \ .$$

Now consider the following variant of RSA in which the quantity $\phi(n) = (p-1)(q-1)$ in the key generation procedure is replaced by $\operatorname{lcm}(p-1, q-1)$.

Set-up: In order to generate their public/private key pair, each user does the following.

1) Generates two large distinct primes p, q.
2) Computes $n = pq$ and $l = \operatorname{lcm}(p-1, q-1)$.
3) Selects an element $e \in \mathbb{Z}_l^*$.
4) Solves the congruence $ed \equiv 1 \pmod{l}$ for $d \in \mathbb{Z}_l^*$.

The user's public and private keys are (e, n) and d, respectively.

Encryption and decryption proceed exactly as in ordinary RSA. That is, plaintexts and ciphertexts are elements in $\mathbb{Z}_n^*$, the encryption of a plaintext $M \in \mathbb{Z}_n^*$ is $C \equiv M^e \pmod{n}$, and the decryption of a ciphertext $C \in \mathbb{Z}_n^*$ is $M \equiv C^d \pmod{n}$.

a. (6 marks) Formally prove that this variant of RSA works. Specifically, prove that

$$(M^e)^d \equiv M \pmod{n}$$

for all $M \in \mathbb{Z}_n^*$.

Solution. Since $ed \equiv 1 \pmod{l}$, there exists $k \in \mathbb{Z}$ such that $ed = 1 + kl$. Moreover, by property (A) of the least common multiple, there exist integers u, v such that $l = u(p-1) = v(q-1)$.

Now let $M \in \mathbb{Z}_n^*$. Then

$$(M^e)^d = M^{ed} = M^{1+kl} = M^{1+ku(p-1)} = M \cdot (M^{p-1})^{ku} \ .$$

By Fermat's Little Theorem, we thus have

$$(M^e)^d \equiv M \cdot (M^{p-1})^{ku} \equiv M \cdot 1^{ku} \equiv M \pmod{p} \ .$$

Similarly, we have

$$(M^e)^d = M^{ed} = M^{1+kl} = M^{1+kv(q-1)} = M \cdot (M^{q-1})^{kv} \ .$$

By Fermat's Little Theorem, we thus have

$$(M^e)^d \equiv M \cdot (M^{q-1})^{kv} \equiv M \cdot 1^{kv} \equiv M \pmod{q} \ .$$

It follows that $M^{ed} \equiv 1 \pmod{p}$ and $M^{ed} \equiv 1 \pmod{q}$. If you are familiar with the Chinese Remainder Theorem, you can use it to conclude that $M^{ed} \equiv 1 \pmod{n}$ as claimed.

If you are unfamiliar with the Chinese Remainder Theorem, you can argue as follows. The congruences $M^{ed} \equiv M \pmod{p}$ and $M^{ed} \equiv M \pmod{q}$ imply that p and q both divide $M^{ed} - M$. By property (B) of the least common multiple applied to $a = p$, $b = q$ and $L = M^{ed} - M$, you can conclude that $l = \text{lcm}(p, q)$ divides $M^{ed} - M$. Now $pq = \text{lcm}(p, q) \cdot \gcd(p, q)$, so since $\gcd(p, q) = 1$, this identity becomes $n = l \cdot 1 = l$. It follows that n divides $M^{ed} - M$ and hence $M^{ed} \equiv M \pmod{n}$ as asserted.

b. (4 marks) Suppose p and q are safe primes, so $p = 2p' + 1$ and $q = 2q' + 1$ with primes p', q'. Formally prove how an attacker who knows n and $l = \text{lcm}(p - 1, q - 1)$ can find factor n, i.e. find p and q.

Solution. Since $\gcd(p', q') = 1$, we have

$$l = \text{lcm}(p - 1, q - 1) = \text{lcm}(2p', 2q') = \frac{(2p')(2q')}{\gcd(2p', 2q')} = \frac{4p'q'}{2\gcd(p'q')} = \frac{4p'q'}{2} = 2p'q' = \frac{\phi(n)}{2} \ .$$

It follows that

$$2l = \phi(n) = (p - 1)(q - 1) = pq - p - q + 1 = n - p - q + 1 \ ,$$

so $p + q = n + 1 - 2l$. Now consider the polynomial

$$f(x) = (x - p)(x - q) = x^2 - (p + q)x + pq = x^2 - (n + 1 - 2l)x + n \ .$$

An attacker with knowledge of n and l knows the coefficients of $f(x)$ and can hence apply the quadratic formula to find the roots p and q of $f(x)$.

Problem 5 — A probabilistic encryption scheme (28 marks)

In this problem, you will investigate a *homomorphic* probabilistic public key cryptosystem.

Set-up: In order to generate their public/private key pair, each user does the following.

1) Generates two large distinct primes p, q such that p does not divide $q-1$ and q does not divide $p-1$;
2) Computes $n = pq$ and $\phi(n) = (p-1)(q-1)$.

The user's public and private keys are n and $\phi(n)$, respectively. The user also precomputes the modular inverse of $\phi(n)$ modulo n, i.e. the element $f \in \mathbb{Z}_n^*$ such that

$$f\phi(n) \equiv 1 \pmod{n} .$$

Plaintexts are elements in $\mathbb{Z}_n$ and ciphertexts are elements in the set

$$Z_{n^2}^* = \{a \in \mathbb{Z} \mid 0 \le a < n^2 \text{ and } \gcd(a, n^2) = 1\} .$$

Encryption: To encrypt a plaintext $M \in \mathbb{Z}_n$, the sender does the following.

1) Generates a random element $r \in \mathbb{Z}_{n^2}^*$.
2) Computes and sends the ciphertext $C \equiv (n+1)^M r^n \pmod{n^2}$.

Decryption: To decrypt a ciphertext $C \in \mathbb{Z}_{n^2}^*$, the receiver does the following.

1) Computes $a \equiv C^{\phi(n)} \pmod{n^2}$.
2) Computes the integer $b = (a-1)/n$.
3) Computes the plaintext $M \equiv bf \pmod{n}$.

a. In this part, you will prove some mathematical preliminaries that are required for the proof that this cryptosystem works.

(i) (2 marks) Using the properties of Euler's phi function covered in class, prove that $\phi(n^2) = n\phi(n)$.

Solution.

$$\phi(n^2) = \phi(p^2q^2) = \phi(p^2)\phi(q^2) = p(p-1)q(q-1) = pq(p-1)(q-1) = n\phi(n) .$$

(ii) (3 marks) Prove that $\gcd(n, \phi(n)) = 1$. This establishes that $\phi(n)$ has an inverse modulo n, so the quantity f defined above exists.

Solution. The only divisors of n are 1, p, q and n. Since $\phi(n) < n$, the only candidates for common divisors of n and $\phi(n)$ are 1, p and q, so it suffices to show that neither p nor q divides $\phi(n)$. Now clearly p does not divide $p-1$ and by assumption, p does not divide $q-1$, so p does not divide $(p-1)(q-1) = \phi(n)$. Similarly, q does not divide $q-1$ and q does not divide $p-1$ by assumption, so q does not divide $\phi(n)$. It follows that $\gcd(n, \phi(n)) = 1$.

(iii) (2 marks) Prove that the number b in step 2 of the decryption procedure is an integer.

Solution. Since n divides n^2, the congruence $a \equiv C^{\phi(n)} \pmod{n^2}$ in step 1 of decryption implies $a \equiv C^{\phi(n)} \pmod{n}$. By Euler's Theorem, $C^{\phi(n)} \equiv 1 \pmod{n}$, so $a \equiv 1 \pmod{n}$. It follows that $a - 1$ is divisible by n, so $b = (a-1)/n$ is an integer.

(iv) (3 marks) Prove that $(n+1)^k \equiv kn + 1 \pmod{n^2}$ for all positive integers k.

Solution. By the binomial theorem, we have

$$(n+1)^k = \sum_{i=0}^{k} \binom{n+1}{i} n^i = n^k + kn^{k-1} + \cdots + \binom{n}{2} n^2 + kn + 1 \ .$$

Every summand except for the last two is divisible by n^2, so $(n+1)^k \equiv kn+1 \pmod{n^2}$.

b. (6 marks) Prove correctness of this cryptosystem. Specifically, prove that if C is the encryption of any plaintext $M \in \mathbb{Z}_n$, obtained via the encryption procedure above, then applying the decryption procedure above to C yields M.

Solution. From step 1 of decryption, step 2 of encryption and part (a) (iv), we have

$$a \equiv C^{\phi n} \equiv (n+1)^{M\phi(n)} r^{n\phi(n)} \equiv (M\phi(n)n + 1) r^{n\phi(n)} \pmod{n^2} \ .$$

Now $n\phi(n) = \phi(n^2)$ by part (a) (i), so $r^{n\phi(n)} \equiv r^{\phi(n^2)} \equiv 1 \pmod{n^2}$ by Euler's Theorem. Hence

$$a \equiv M\phi(n)n + 1 \pmod{n^2} \ ,$$

or equivalently, there exists $k \in \mathbb{Z}$ sich that

$$a = M\phi(n)n + 1 + kn^2 \ .$$

It follows that $b = \dfrac{a-1}{n} = M\phi(n) + kn$, or equivalently

$$b \equiv M\phi(n) \pmod{n} \ .$$

Since $f\phi(n) \equiv 1 \pmod{n}$, it follows that

$$bf \equiv M\phi(n)f \equiv M \pmod{n} \ ,$$

so the decryption procedure produces the plaintext M.

c. (3 marks) For any plaintext $M \in \mathbb{Z}_n$, denote by $E(M, r)$ the encryption of M under public key n using the random number r for encryption. Prove that for any two plaintexts $M_1, M_2 \in \mathbb{Z}_n$, any elements $r_1, r_2 \in \mathbb{Z}_n^*$, and any positive integer k, we have

$$E(M_1, r_1) \cdot E(M_2, r_2) = E(M_1 + M_2, r_1 r_2) \ \text{ and } \ E(M_1, r_1)^k = E(kM_1, r_1^k) \ .$$

In other words, if we apply decryption to the above identities, we see that decrypting a product of two encryptions yields the sum of the plaintexts, and decrypting a power of an encryption yields the corresponding multiple of the plaintext.[3]

[3] Cryptosytems satisfying these two properties are called *linearly homomorphic*. The first property in particular makes this system very suitable for electronic voting, especially for yes/no elections. Every user encrypts their vote, which is 1 for "yes" and 0 for "no", using the election administrator's public key n. The administrator multiplies all the encrypted votes together and decrypts the product. By the first property, this decryption is the sum of all the votes, which is just the number of "yes" votes (note that the administrator need not know the random numbers used by the voters to encrypt their votes). This voting procedure preserves secrecy of individual votes and anonymity of voters. Moreover, the election administrator only needs to compute a modular product and perform one decryption, which is much more efficient than decrypting each vote individually.

Solution. We have $E(M_1, r_1) \equiv (n+1)^{M_1} r_1^n \pmod{n}$ and $E(M_2, r_2) \equiv (n+1)^{M_2} r_2^n \pmod{n}$ with $r_1, r_2 \in \mathbb{Z}^*_{n*2}$. Thus,

$$\begin{aligned} E(M_1, r_1) \cdot E(M_2, r_2) &\equiv \Big((n+1)^{M_1} r_1^n\Big)\Big(n+1)^{M_2} r_2^n\Big) \\ &\equiv (n+1)^{M_1+M_2} (r_1 r_2)^n \equiv E(M_1 + M_2, r_1 r_2) \pmod{n^2} \end{aligned}$$

and

$$E(M_1, r_1)^k \equiv \Big((n+1)^{M_1} r_1^n\Big)^k \equiv (n+1)^{kM_1} (r_1^k)^n \equiv E(kM_1, r_1^k) \pmod{n^2} \ .$$

Since all residues modulo n^2 are taken to be between 0 and $n^2 - 1$, the above congruences are in fact equalities.

d. Consider the following chosen ciphertext attack on the above system where an attacker wishes to obtain the decryption M of a ciphertext C and proceeds as follows.

1) Generates $r \in \mathbb{Z}^*_{n^2}$ with $r^n \not\equiv 1 \pmod{n^2}$.
2) Computes $C' \equiv Cr^n \pmod{n^2}$ (this is the chosen ciphertext).
3) Obtains the decrytion M' of C'.

(i) (2 marks) Prove that $C' \neq C$, so C' is a valid choice of ciphertext to get decrypted in step 3.

Solution. We have $C' = C$ if and only if $C \equiv Cr^n \pmod{n^2}$. Since $\gcd(C, n^2) = 1$, C has an inverse modulo n^2, and multiplication by this inverse yields $C' = C$ if and only if $r^n \equiv 1 \pmod{n^2}$. But this was explicitly excluded in step 1 of the CCA, so $C' \neq C$.

(ii) (3 marks) Prove that $M' = M$. In other words, the decryption obtained in step 3 of the CCA is precisely the plaintext that the attacker is after, and thus the attack is successful.

Solution. In step 1 of decryption, we have

$$(C')^{\phi(n)} \equiv (Cr^n)^{\phi(n)} \equiv C^{\phi(n)} r^{n\phi(n)} \pmod{n^2} \ .$$

By part (a) (i), $n\phi(n) = \phi(n^2)$, and Euler's Theorem thus yields $r^{n\phi(n)} \equiv 1 \pmod{n^2}$. It follows that $(C')^{\phi(n)} \equiv C^{\phi(n)} \pmod{n^2}$, so the element a computed in step 1 of decryption is the same when decrypting C' and C. Thus, the integer b in step 2 of decryption is also identical for C' and C, hence so is the plaintext obtained in step 3 of decryption.

e. (4 marks) An element $z \in \mathbb{Z}^*_{n^2}$ is an *n-th power residue modulo n^2* if and only if there exists an element $x \in \mathbb{Z}_{n^2}$ such that $x^n \equiv z \pmod{n^2}$. Prove that a ciphertext C is the encryption of the plaintext $M = 0$ if and only if C is an n-th power residue modulo n^2. So decryption for this cryptosystem can be thought of as an "n-th power residue modulo n^2 detector".[4]

Solution. The encryption of $M = 0$ is $C \equiv (n+1)^0 r^n \equiv r^n \pmod{n^2}$ which is an n-th power residue modulo n^2.

[4] Detecting n-th power residues modulo n^2 is generally believed to be computationally intractable when n is the product of two distinct large primes.

Conversely, suppose C is an n-th power residue modulo n^2; say, $C \equiv x^n \pmod{n^2}$ for some $x \in \mathbb{Z}_{n^2}^*$. Then step 1 of decryption yields

$$a \equiv C^{\phi(n)} \equiv x^{n\phi(n)} \equiv x^{\phi(n^2)} \equiv 1 \pmod{n^2}$$

by part (a) (i) and Euler's Theorem. It follows that $a = 1$ in step 1 of decryption, and hence $b = 0$ in step 2 of decryption. Finally, this forces $m \equiv 0 \cdot f \equiv 0 \pmod{n}$ in step 3 of decryption, so the decryption of C is $M = 0$. Equivalently, C is the encryption of $M = 0$.

Remark: The system above is known as the *Paillier cryptosystem*, after Pascal Pallier who invented it in 1999 as the first linearly homomorphic cryptosystem.

Written Problem for MATH 318 only

Problem 6 — An RSA-like public key cryptosystem (38 marks)

In this problem, you will investigate a public key cryptosystem that is similar to RSA but has the advantage that an adversary is able to break the system *if and only if* she is able to factor the modulus. The price to pay for this potential added security is the restriction on the format of plaintexts: they must have Jacobi symbol 1 with respect to n. In addition, the system is extremely vulnerable to a simple chosen ciphertext attack that does not work for RSA.

In part(c), we will extend the plaintext space to be all of $\mathbb{Z}_n^*$ (like RSA), but at the additional cost of more complicated encryption and decryrption procedures as well as reduced efficiency: in addition to a modular exponentiation (like in RSA), encryption in the extended scheme requires the evaluation of a Jacobi symbol.

Set-up: In order to generate their public/private key pair, each user does the following.

1) Generates two large primes p and q with $p \equiv 3 \pmod 4$ and $q \equiv 3 \pmod 4$.
2) Computes $n = pq$ and $f = \phi(n)/4$.
3) Selects $e \in \mathbb{Z}$ with $1 < e < f$ and $\gcd(2e, f) = 1$
4) Solves the congruence $2ed \equiv 1 \mod f$ for $d \in \mathbb{Z}$, $1 \leq d < f$.

The user's public and private keys are (e, n) and d, respectively. Here, as always, $\phi()$ denotes the Euler phi function.

The plaintext space is the set of all $M \in \mathbb{Z}_n^*$ such than $\left(\dfrac{M}{n}\right) = 1$. Encryption of such a plaintext M with public key (e, n) is given by

$$C \equiv M^{2e} \pmod n \ .$$

Decryption of a ciphertext C is given as

$$M \equiv C^d \pmod n \ .$$

Unfortunately, the 2 in the encryption exponent introduces the same abiguity as squaring of ordinary integers. Specifically, it makes it impossible to distinguish M from $n - M$ (which is just $-M \pmod n$) after decryption. Now that one way to distinguish these two numbers is by their parity: one of them is odd and the other is even (since n is odd). So the encrypter simply needs to send the parity bit distinguishing M from $n-M$, along with the ciphertext. Specifically, encryption and decryption proceed as follows:

Encryption: To encrypt a plaintext $M \in \mathbb{Z}_n^*$ with $\left(\frac{M}{n}\right) = 1$, the sender does the following.

1) If M is even, put $b = 0$, else put $b = 1$.
2) Compute $C \equiv M^{2e} \pmod n$.
3) Send (C, b).

Decryption: To decrypt a pair (C, b) , the receiver does the following.

1) Compute $M' \equiv C^d \pmod n$.
2) If M' is even and $b = 0$ or M' is odd and $b = 1$, put $M = M'$, else put $M = n - M'$.

Remark: The restrictions $\left(\frac{M}{n}\right) = 1$ and $p \equiv q \equiv 3 \pmod 4$ ensure the condition

$$\left(\frac{M}{p}\right)^{(q-1)/2} = \left(\frac{M}{q}\right)^{(p-1)/2} \tag{1}$$

for all $M \in \mathbb{Z}_n^*$ which is crucial for this cryptosystem to work; you will use this in part (a) (iii).

a. (Correctness, 14 marks)

(i) (2 marks) Prove that $(p-1)/2$ and $(q-1)/2$ are odd.

Solution. Write $p = 4k+3$ for some positive integer k. Then $(p-1)/2 = 2k+1$ is odd; similarly, $(q-1)/2$ is odd.

(ii) (3 marks) Prove that $\left(\frac{M}{p}\right) = \left(\frac{M}{q}\right)$ for all $M \in \mathbb{Z}_n^*$ with $\left(\frac{M}{n}\right) = 1$. Then use part (a) (i) to conclude that property (1) above holds.

Solution. Let $M \in \mathbb{Z}_n^*$ with $\left(\frac{M}{n}\right) = 1$. Then $1 = \left(\frac{M}{n}\right) = \left(\frac{M}{p}\right)\left(\frac{M}{q}\right)$. Since $\left(\frac{M}{p}\right)$ and $\left(\frac{M}{q}\right)$ take on values in $\{1, -1\}$ and their product is 1, they must be equal (i.e. both are 1 or both are -1).

Now $(-1)^m = -1$ for every odd integer m, and of course $1^m = 1$ for all integers m. It follows that

$$\left(\frac{M}{p}\right)^{(q-1)/2} = \left(\frac{M}{p}\right) = \left(\frac{M}{q}\right) = \left(\frac{M}{q}\right)^{(p-1)/2} .$$

(iii) (4 marks) Prove that $M^f \equiv \pm 1 \pmod n$ for all $M \in \mathbb{Z}_n^*$ with $\left(\frac{M}{n}\right) = 1$.

Hint: Specifically, prove $M^f \equiv e \pmod n$ where $e = \left(\frac{M}{p}\right)^{(q-1)/2} = \left(\frac{M}{q}\right)^{(p-1)/2}$ is the quantity in (1).

Solution. Let $M \in \mathbb{Z}_n^*$ with $\left(\frac{M}{n}\right) = 1$. Then

$$M^f = M^{(p-1)(q-1)/4} = \left(M^{(p-1)/2}\right)^{(q-1)/2} = \left(M^{(q-1)/2}\right)^{(p-1)/2} .$$

By Euler's criterion, we have

$$(M^{(p-1)/2})^{(q-1)/2} \equiv \left(\frac{M}{p}\right)^{(q-1)/2} \equiv e \pmod p$$

and

$$(M^{(q-1)/2})^{(p-1)/2} \equiv \left(\frac{M}{q}\right)^{(p-1)/2} \equiv e \pmod q .$$

It follows that p and q both divide $M^f - e$; hence n must also divide $M^f - e$ (analogous reasoning as was used in Problem 4 (a)). Hence $M^f \equiv e \equiv \pm 1 \pmod n$.

(iv) (5 marks) Use part (a) (iii) to prove correctness of this cryptosystem. Specifically, prove that if C is the encryption of any plaintext $M \in \mathbb{Z}_n^*$ with $\left(\frac{M}{n}\right) = 1$, obtained via the encryption procedure above, then applying the decryption procedure above to C yields M.

Solution. This proof is similar to the proof of correctness of RSA decryption. Since $2ed \equiv 1 \pmod{f}$, there exists an integer k such that $2ed = kf + 1$. Note that k is odd. Let $M \in \mathbb{Z}_n*$ with $\left(\frac{M}{n}\right) = 1$. Then

$$C^d \equiv (M^{2e})^d \equiv M^{2ed} \equiv M^{kf+1} \equiv (M^f)^k \cdot M \equiv e^k M \equiv eM \equiv \pm M \pmod{n} \ ,$$

where e is the quantity of (1) (note that $e^k = e$ as k is odd and $e = \pm 1$). Hence $M' = M$ or $M' = n - M$.

If M is even, then $b = 0$. If M' is also even, then step 2 of the decryption procedure correctly identifies M' as M, whereas if M' is odd, then $n - M$ is even and M is correctly identified as $n - M$. Similarly, if M is odd, then $b = 1$. If M' is also odd, then step 2 of the decryption procedure correctly identifies M' as M, whereas if M' is evend, then $n - M$ is odd and M is correctly identified as $n - M$.

b. (Security, 12 marks)

Just as for RSA, it is evident that if an attacker can factor n, i.e. find the primes p and q, she can proceed as the designer and compute f and the private key d, thereby breaking the scheme. In this part, you will prove the "converse": if an attacker can break the scheme via a successful chosen ciphertext attack, she can factor n.

Consider the following chosen ciphertext attack on the above system where an attacker attempts to factor n.

1) Chooses any $y \in \mathbb{Z}_n^*$ such that $\left(\dfrac{y}{n}\right) = -1$.
2) Computes $C \equiv y^2 \pmod{n}$ (this is the chosen ciphertext).
3) Obtains the decryption M of C.
4) Computes $x \equiv M^e \pmod{n}$.
5) Computes $\gcd(x - y, n)$.

(i) (2 marks) Prove that $\left(\dfrac{-1}{n}\right) = 1$.

Solution. Since $(p-1)/2$ and $(q-1)/2$ are both odd, we have $\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2} = -1$ and $\left(\frac{-1}{q}\right) = (-1)^{(q-1)/2} = -1$, so $\left(\frac{-1}{n}\right) = (-1)(-1) = 1$.

(ii) (2 marks) Prove that $\left(\dfrac{M}{n}\right) = 1$ and hence $\left(\dfrac{x}{n}\right) = 1$, with M as given in step 3 and x as in step 4 of the CCA.

Solution. Since M is the decryption of C, we have $M \equiv C^d \equiv (y^2)^d \equiv y^{2d} \pmod{n}$, so $Mn = \left(\dfrac{y}{n}\right)^{2d} = 1$ and hence $\left(\dfrac{x}{n}\right) = \left(\dfrac{M}{n}\right)^e = 1^e = 1$.

(iii) (2 marks) Prove that $x^2 \equiv y^2 \pmod{n}$.

Solution. Since M is the decryption of C, we have $C \equiv M^{2e} \pmod{n}$, so

$$x^2 \equiv (M^e)^2 \equiv M^{2e} \equiv C \equiv y^2 \pmod{n} \ .$$

(iv) (3 marks) Use parts (b) (i)-(ii) to prove that $x \not\equiv y \pmod{n}$ and $x \not\equiv -y \pmod{n}$.
Hint: Compare $\left(\dfrac{x}{n}\right)$ with $\left(\dfrac{y}{n}\right)$ and $\left(\dfrac{-y}{n}\right)$.

Solution. We have $\left(\dfrac{y}{n}\right) = -1$ by step 1 of the attack and hence

$$\left(\frac{-y}{n}\right) = \left(\frac{-1}{n}\right)\left(\frac{y}{n}\right) = 1 \cdot (-1) = -1$$

by part (b) (i). Also $\left(\dfrac{x}{n}\right) = 1$ by part (b) (ii), so

$$\left(\frac{x}{n}\right) \neq \left(\frac{y}{n}\right) \text{ and } \left(\frac{x}{n}\right) \neq \left(\frac{-y}{n}\right) .$$

Now if $a \equiv b \pmod{n}$, then $\left(\dfrac{a}{n}\right) = \left(\dfrac{b}{n}\right)$ for all $a, b \in \mathbb{Z}_n^*$. Thus $x \not\equiv y \pmod{n}$ and $x \not\equiv -y \pmod{n}$.

(v) (3 marks) Finally, the grand finale: use parts (b) (ii)-(iv) to prove that $\gcd(x-y, n) = p$ or q, so the attack above factors n.

Solution. Note that n divides $x^2 - y^2$ by part (b) (iii), so n divides $(x-y)(x+y)$. For brevity, put $g = \gcd(x-y, n)$. Then $g \in \{1, p, q, n\}$. We need to rule out the cases $g = n$ and $g = 1$.

If $g = n$, then n divides $x - y$, so $x \equiv y \pmod{n}$, contradicting part (b) (iv).

Ife $g = 1$, then n is coprime to $x - y$ and must hence divide $x + y$. But then $x \equiv -y \pmod{n}$, again contradicting part (b) (iv).
It follows that $g = p$ or q.

c. (Expanding the plaintext space to all of $\mathbb{Z}_n^*$, 12 marks)

The idea behind this part is to choose the primes p and q such that for any $M \in \mathbb{Z}_n^*$, if $\left(\frac{M}{n}\right) = -1$ (so M is not a valid plaintext) then $\left(\frac{2M}{n}\right) = 1$ (so $2M \pmod{n}$ is a valid plaintext). In addition to sending along a parity bit, the encrypter then also needs to send another bit indicating whether $\left(\frac{M}{n}\right) = -1$ (in which case $2M$ is encrypted instead of M). You will prove in part (c) (ii) that an appropriate choice is

$$p \equiv 3 \pmod{8} , \qquad q \equiv 7 \pmod{8} ,$$

so we assume these congruence conditions from here on.

Consider the following modified encryption procedure, where the plaintext space is now all of $\mathbb{Z}_n^*$:

Encryption: to encrypt a plaintext $M \in \mathbb{Z}_n^*$, proceed as follows.

1) Compute $\left(\frac{M}{n}\right)$.
2) If $\left(\frac{M}{n}\right) = 1$, put $M_0 = M$ and $b_0 = 0$, else put $M_0 \equiv 2M \pmod{n}$ and $b_0 = 1$.
3) If M_0 is even, put $b_1 = 0$, else put $b_1 = 1$.
2) Compute $C \equiv M_0^{2e} \pmod{n}$.
3) Send (C, b_0, b_1).

Decryption: to decrypt a triple (C, b_0, b_1), proceed as follows.

1) Compute $L \equiv C^d \pmod{n}$.
2) If L is even and $b_1 = 0$ or L is odd and $b_1 = 1$, put $L_0 = L$, else put $L_0 = n - L$.
3) If $b_0 = 0$, put $M = L_0$, else put $M \equiv \frac{n+1}{2} L_0 \pmod{n}$

Note that the quantity $(n+1)/2$ in step 3 of decryption is simply the inverse of 2 modulo n as $1 \leq (n+1)/2 < n$ and $2 \cdot (n+1)/2 = n + 1 \equiv 1 \pmod{n}$.

(i) (3 marks) Prove that $\left(\dfrac{2}{n}\right) = -1$.

Remark: This shows that $y = 2$ is a valid choice in step 1 of the CCA above.

Solution. Recall that $\left(\frac{2}{N}\right) = (-1)^{(N^2-1}8$ for any odd integer $N \geq 3$. Write $p = 3 + 8k$ and $q = 7 + 8m$ for some integers k, m. Then

$$\frac{p^2 - 1}{8} = \frac{1}{8}(8 + 48k + 64k^2) = 1 + 6k + 8k^2$$

and

$$\frac{q^2 - 1}{8} = \frac{1}{8}(48 + 112m + 64m^2) = 6 + 14m + 8m^2 \ ,$$

so $(p^2 - 1)/8$ is odd and $(q^2 - 1)/2$ is even. It follows that $\left(\dfrac{2}{p}\right) = (-1)^{(p^2-1)/8} = -1$ and $\left(\dfrac{2}{q}\right) = (-1)^{(q^{-}1)/2} = 1$, so $\left(\frac{2}{n}\right) = \left(\frac{2}{p}\right)\left(\frac{2}{q}\right) = -1$.

(ii) (2 marks) Prove that $\left(\frac{M_0}{n}\right) = 1$, with M_0 as in step 2 of the modified encryption procedure.

Solution. If $\left(\frac{M}{n}\right) = 1$, then $M_0 = M$, so $\left(\frac{M_0}{n}\right) = \left(\frac{M}{n}\right) = 1$.
If $\left(\frac{M}{n}\right) = -1$, then $M_0 \equiv 2M \pmod{n}$, so $\left(\frac{M_0}{n}\right) = \left(\frac{2}{n}\right)\left(\frac{M}{n}\right) = (-1)(-1) = 1$ by part (c) (i).

(iii) (4 marks) Prove that $L_0 = M_0$, with L as in step 2 of decryption and M_0 as in step 2 of decryption.

Solution. Using the same reasoning as in part (a) (iii), we obtain

$$L \equiv C^d \equiv (M_0^{2e})^d \equiv M_0^{2ed} \equiv \pm M_0 \pmod{n} \ .$$

Since $L_0 = L$ or $L_0 = n - L$, we see that $L_0 \equiv \pm L \pmod{n}$, so $L_0 \equiv \pm M_0 \pmod{n}$. Hence $L_0 = M_0$ or $L_0 = n - M_0$. Recall again that M_0 and $n - M_0$ have different parity (i.e. one of them is odd and the other is even); similarly for L_0 and $n - L_0$.

Suppose first that $b_1 = 0$, so M_0 is even and $n - M_0$ is odd. If L is even, then step 3 of decryption defines $L_0 = L$, so L_0 is even, forcing $L_0 = M_0$. If L is odd, then $L_0 = n - L$ in step 3 of decryption, which is even, so again $L_0 = M_0$.

Suppose now that $b_1 = 1$, so M_0 is odd and $n - M_0$ is even. If L is even, then step 3 of decryption defines $L_0 = n - L$ which is odd, forcing $L_0 = M_0$. If L is odd, then $L_0 = L$ in step 3 of decryption, which is odd, so again $L_0 = M_0$.

(iv) (3 marks) Use part (c) (iii) to prove correctness of this modified version of this cryptosystem. Specifically, prove that if C is the encryption of any plaintext $M \in \mathbb{Z}_n*$, obtained via the modified encryption procedure above, then applying the modified decryption procedure above to C yields M.

Solution. By part (c) (iii), we have $L_0 = M_0$, so we only need to show that step 3 of decryption produces the plaintext that was encrypted to obtain C. If $b_0 = 0$, then step 2 of encryption defined M_0 to be M, so we have $M = L_0 = M_0$ in step 3 of decryption which is correct. If $b_0 = 1$, then step 2 of encryption defined M_0 to be $2M \pmod{n}$, so

$$\frac{n+1}{2} L_0 \equiv \frac{n+1}{2} M_0 \equiv \frac{n+1}{2}(2M) \equiv \left(\frac{n+1}{2} \cdot 2\right) M \equiv (n+1)M \equiv M \pmod{n} .$$

The scheme in thus problem is known as the *Rabin-Williams* cryptosystem, after its inventors. In 1979, Michael Rabin introduced the idea of squaring in the context of a signature scheme, which an adversary can break, i.e. forge signatures via a chosen message attack, if and only if she can factor the modulus. In 1980, Hugh C. Williams, who is a former math professor at U Calgary (and long retired by now) proposed the public key cryptosystem above which merges Rabin's squaring approach with RSA.

Programming Problem for CPSC 418 only

Problem 7 — Vaccine Passports (38 marks)

For more information on SRP, see Thomas Wu's website *The Secure Remote Password Protocol*, http://srp.stanford.edu/, Stanford University.

More information about cipher suites can be found at https://www.gnutls.org/manual/html_node/Supported-ciphersuites.html.

Working code for this problem is posted separately on the Piazza resources page https://piazza.com/ucalgary.ca/fall2021/cpsc418math318/resources.