ITIS 6200/8200 Principles of Information Security and Privacy

Midterm

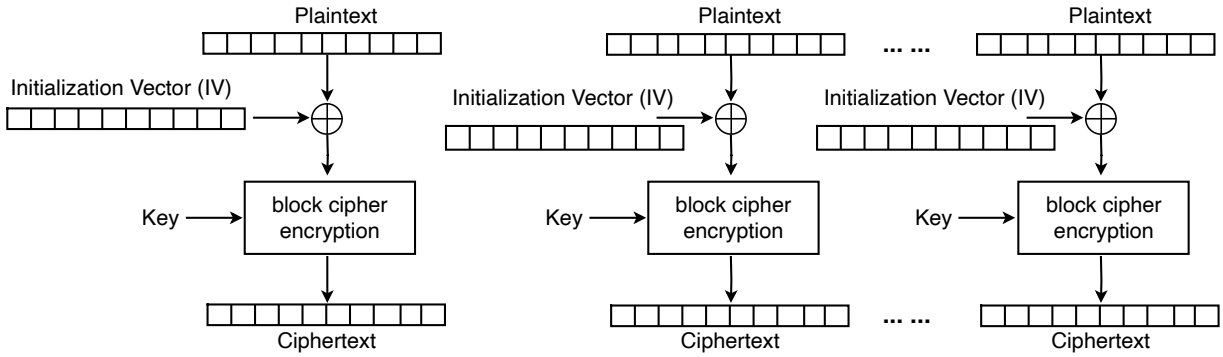
Question 1. Security Principle (10 points)

We have learned 11 security principles in the class. Name two of them and give real-life examples. Don't use examples from our slides and assignment #2.

Grading Notes: The answers may vary, as long as they are reasonable and the example matches the security principle. 5 points for each principle.

Question 2. **Block Cipher Design (15 points)**

One student in our class suggests the following design of block cipher operating mode. The same IV is used for every block. The message M is split into j plaintext blocks M_1 … M_j each of size *n*. The encryption mode outputs (IV, C_1, …, C_j) as the overall ciphertext. Assume that IV is randomly generated per encryption.



Q (a): Write down the encryption formula. That is, what is the formula for C_i ($0 < i <= j$) given (1) plaintext M_1 … M_j (2) encryption algorithm Enc(K, M) which takes a key K and message M as inputs, and (3) a randomly generated IV. (5 points)

Ans: Ci = Enc(Key, Mi ⊕ IV**)**

Grading Notes: Give partial points if not correct: 1 point if Enc() is used; 1 point if ⊕ IV is used.

Q (b): Write the decryption formula for M_i ($0 < i <= j$) using this mode. That is, how to get M_1 and M_i ($0 < i <= j$) given (1) ciphertext (IV, C_1, …, C_j) and (2) decryption Dec(K, M). (5 points)

Ans: Mi = Dec(Key, Ci**)** ⊕ IV

Grading Notes: I had a typo in the original midterm, though it should not matter. **We shall give 5 points to all students.**

Q (c): Is this mode IND-CPA secure? If yes, explain why; if not, describe how an attacker can break IND-CPA. (Hint: find two example messages that if Eve sends to Alice for encryption and Alice randomly chooses one for encryption and sends back to Eve. Then Eve can tell which message Alice encrypts with a probability > 0.5) (5 points)

Ans: Eve can send two messages: Ma and Mb, to Alice for encryption. Ma has the same plaintext for all blocks; and Mb has different plaintext for each block. If the ciphertext returned by Alice shows repeated patterns, it is Ma being encrypted. Otherwise, Mb.

Grading Notes: Give 2 points if the student mentions frequency attack, but no explanation.

Question 3. **Hashing (10 points)**

Alice's computer stores the files in the following way: for every file F, the computer will calculate the hash value of the file hash(F) and store it after the file, i.e., F || hash(F). Every time when Alice login, the machine will automatically hash all the files and compare the results to the stored hash values. In this way, if by accident the hard drive is mis-functioning and flips a few bits in a file, Alice can immediately detect it since the hash value will be different. Now an attacker hacks into Alice's machine and he tries to change several files. The attacker also knows the hash function that the computer uses, e.g., SHA256.

Q 3.1: Describe what the attacker needs to do so that the next time Alice login, the machine will not detect the changes. (5 points)
Ans: Change the F into F', and then append hash(F'), that is, F' || hash(F')

Q 3.2: How do we improve the mechanism to prevent / detect such changes? (5 points)
Ans: Use a keyed hash, e.g., HMAC(k, F) with Alice's private key.

Grading notes: there may be other approaches.

Question 4. **Diffie-Hellman Key Exchange (15 points)**

In Diffie-Hellman key exchange, there are values of a, b, g and p. Alice computes $g^a \bmod p$ and sends it to Bob; Bob computes $g^b \bmod p$ and sends it to Alice. Then Alice computes $(g^b \bmod p)^a \bmod p$, and Bob computes $(g^a \bmod p)^b \bmod p$.

Q 4.1: Which of these values (a, b, g, and p) are publicly known and which must be kept private? (5 points)
Ans: g and p are public; a and b are private

Q 4.2: Mallory is powerful attacker that can eavesdrop, intercept, and modify messages sent between Alice and Bob. Alice and Bob use Diffie-Hellman to agree on a shared symmetric key K. After the exchange, Bob feels somethings is wrong and calls Alice. He realizes his K is different from Alice's K. Explain what Mallory has done. (5 points)
Ans: Mallory intercepts Alice's message, creates a m value, and sends $(g^m \bmod p)$ to Alice and Bob. Alice receives $(g^m \bmod p)$ from Mallory, and computes $(g^m \bmod p)^a \bmod p = g^{ma} \bmod p$. Meanwhile, Bob receives $(g^m \bmod p)$ from Mallory, and computes $(g^m \bmod p)^b \bmod p = g^{mb} \bmod p$. Thus, Alice and Bob get different values.

Q 4.3: In Diffie-Hellman key exchange, p should be a large prime. What happens if p is a small number? If the exchanged key is used in symmetric key encryption, how would that affect the security of the encryption? (5 points)
Ans: The number of possible keys would be very small, and if it's used in symmetric key encryption, it is easy for an attacker to brute-force the possible keys.

Question 5. **Public-Key Encryption (15 points)**

Bob has a public-private key pair (pub_Bob, priv_Bob). Alice needs to send some information to Bob. She wants to make sure that when Bob opens the message, he can verify that this is from Alice but not anyone else.

Q 5.1: If Alice sends out the message as: [Alice || E_{pub_Bob}(message)] to Bob. That is, she sends out her name in clear text, followed by ciphertext of the message encrypted with Bob's public key. Can powerful attacker Mallory impersonate Alice and send out a packet in Alice's name? How can she do it? Assume that Mallory also has the public key of Bob. (5 points)
Ans: Yes. Mallory can send Bob [Alice || E_{pub_Bob}(M')], Bob won't notice.

Q 5.2: If Alice sends out [E_{pub_Bob}(Alice || message)], where Alice puts her name in the encryption. Can Mallory still impersonate Alice? (5 points)
Ans: Yes. Mallory can send Bob [E_{pub_Bob}(Alice || M')], Bob won't notice.

Q 5.3: Design a way that Bob can ensure the message comes from Alice. (5 points)
Ans: Use Digital signatures: use Alice's private key for signature.
Grading notes: There may be other approaches.

Question 6. **RSA Signature (15 points)**

To use RSA signatures on messages, we first create a RSA key pair: (N, e) is the RSA public key and d is the RSA private key, where N is the RSA modulus. For standard RSA signatures, we typically set e to a small prime value such as 5; for this problem, let e = 5.

Q 6.1: For RSA signatures, we often sign the hash of a message, rather than the message directly. Why is that? (5 points)
Ans: So the RSA can sign messages of different size.

Q 6.2: Assume that we **skip using a hash function**, and sign the messages directly. That means, if Alice wants to send a signed message to Bob, she will send (M, S) to Bob where $S = M^d \bmod N$ is computed using her private signing key d. With such a scheme, how does Bob verify this message come from Alice? What formula does Bob need? (5 points)
Ans: check if $M = S^e \bmod N$, if yes, then it's from Alice.

Q 6.3: Mallory learns that Alice and Bob are using the simplified signature scheme that without using hash functions. Can Mallory find a (M, S) pair such that S will be a valid signature on M? Assume that Mallory knows Alice's public key N and e, but not Alice's private key d. (5 points)
Ans: M = S = 1 or M = 0 and S = 1.

Question 7. **Confidentiality & Integrity (20 points)**

Alice wants to send messages to Bob in an insecure channel. The attackers may eavesdrop, intercept, and modify the messages sent in the channel. By combining the techniques that we have learned in this course, **develop two schemes that can ensure both integrity and confidentiality** of the communications. Describe how Alice produces the ciphertext and how Bob decrypts the ciphertext to read the message. Given a plaintext M, write down the formula of the ciphertext sent from Alice to Bob, and describe how Bob can decrypt C into M. Explain why the scheme provides both confidentiality and integrity.

You can make the following assumptions:

1. Alice and Bob already shared the secret key if they use symmetric key encryption.
2. Alice already knows Bob's public key, and Bob already knows Alice's public key.
3. Alice and Bob's private keys are not compromised.
4. Every technique is secure with respect to its requirements, e.g., AES uses random IV or nonce.
5. Cryptographic tools do not interfere with each other when used in combination, e.g., the same key can be used for AES and MAC.

Hint: here are some of techniques and notations that may be useful.

- AES: Enc(K, M) and Dec(K, C) where K, M, and C denote the key, plaintext, and ciphertext
- HMAC: HMAC(K, V) where K, V are the inputs to the HMAC function
- Hash: Hash(M) where M is the input to the hash function
- RSA public-key encryption: Enc(pub-Bob, M), and Dec(priv-Bob, C).
- RSA signature: Enc(priv-Alice, M) and Dec(priv-Alice, C)

Q 7.1: Scheme #1 (10 points)

Q 7.2: Scheme #2 (10 points)

Extra credit (5 points)
We have learned other techniques in the course. Can you use these techniques to relax/revise any assumption above?

Answer: At least three schemes are taught in the class.
Scheme #1: AES + MAC
Enc(K, M) || MAC(K, Enc(K, M))

Scheme #2: AES + RSA signature

1. Assume Alice and Bob know each other's public key.

2. Assume Alice and Bob share a secret key K for AES encryption.
3. Alice sends Bob **C = Enc(priv-Alice, Enc(K, M)).**
4. Bob can decrypt with **M = Dec(K, Dec(pub-Alice, C))**

Scheme #3: RSA public-key encryption + Signature

1. Bob produces his key pair: pub-Bob, priv-Bob
2. Certificate authority produces a certificate for Bob's public key.
3. Alice requests Bob's public key from CA, gets Bob's public key.
4. Alice sends Bob **C = Enc(priv-Alice, Enc(pub-Bob, M)).**
5. Bob can decrypt with **M = Dec(private-Bob, Dec(public-Alice, C))**

Extra credits:

1. Use Diffie-Hellman public-key exchange to relax the assumption 1.
2. Use CA to relax the assumption 2 that Alice already knows Bob's public key, and Bob already knows Alice's public key. We can use CA to distribute Alice and Bob's public keys.