ITIS 6200/8200 Principles of Information Security and Privacy

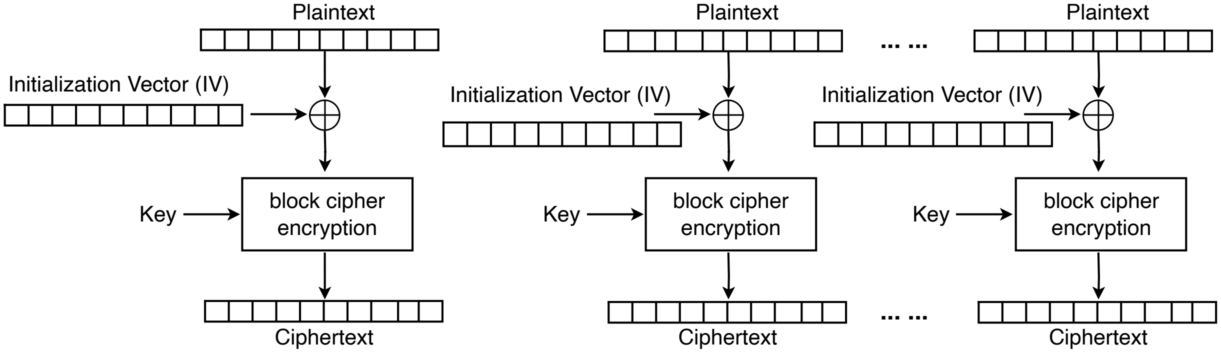
Midterm

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 the examples from our slides and assignments.

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 $M_1 \dots M_j$ each of size *n*. The encryption mode outputs (IV, C_1, …, C_j) as the overall ciphertext. Assume IV is randomly generated per encryption.



Q 2.1: Write down the encryption formula. That is, what is the formula for C_i ($0 < i <= j$) given (1) plaintext $M_1 \dots M_j$ (2) encryption algorithm Enc(K, M) which takes a key K and message M as inputs, and (3) a randomly generated IV.

Q 2.2: 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) encryption algorithm Enc(K, M).

Q 2.3: 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 find a way to tell which message Alice encrypts with a probability > 0.5)

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.

Q 3.2: How do we improve the mechanism to prevent / detect such changes?

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 mod p and sends it to Bob; Bob computes g^b mod p and sends it to Alice. Then Alice computes $(g^b \bmod p)^a$ mod p, and Bob computes $(g^a \bmod p)^b$ mod p.

Q 4.1: Which of these values (a, b, g, and p) are publicly known and which must be kept 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.

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?

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.

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?

Q 5.3: Design a way that Bob can ensure the message comes from Alice.

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?

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?

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.

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

Q 7.2: Scheme #2

Extra credit (5 points)

We have learned other techniques in the course, for example, Diffie-Hellman public-key exchange, Certificate Authority (CA). Can you use these techniques to relax/revise any assumption above? How?