Solution_CS4331/CS5342 Network Security
Homework_2

Q.1. What are the pros and cons of public-key cryptography (4 points)

Ans: Public-key cryptography is a type of encryption that encrypts and decrypts data using two separate keys called public key and Private key.

Pros or Advantages:

• Easy key distribution

• No longer need to assume that Alice and Bob already share a secret.

Cons or Disadvantages:

• Much slower than symmetric-key cryptography

• Number theory calculations are much slower than XORs and bit-shifts.

Q.2. What are the properties of public key encryption? (4 points)

Ans:

• **Correctness**: Decrypting a ciphertext should result in the message that was originally encrypted

Dec(SK, Enc(PK, M)) = M for all PK, SK ← KeyGen() and M

• **Efficiency**: Encryption/decryption should be fast

• **Security**: Alice (the challenger) just gives Eve (the adversary) the public key, and Eve doesn't request encryptions. Eve cannot guess out anything.

Computationally infeasible to recover M with PK and ciphertext.

Q.3. Describe the steps of public key encryption with example (4 points)

Ans: Steps for Public key encryption:

Step1: Generate a pair of keys.

Step2:

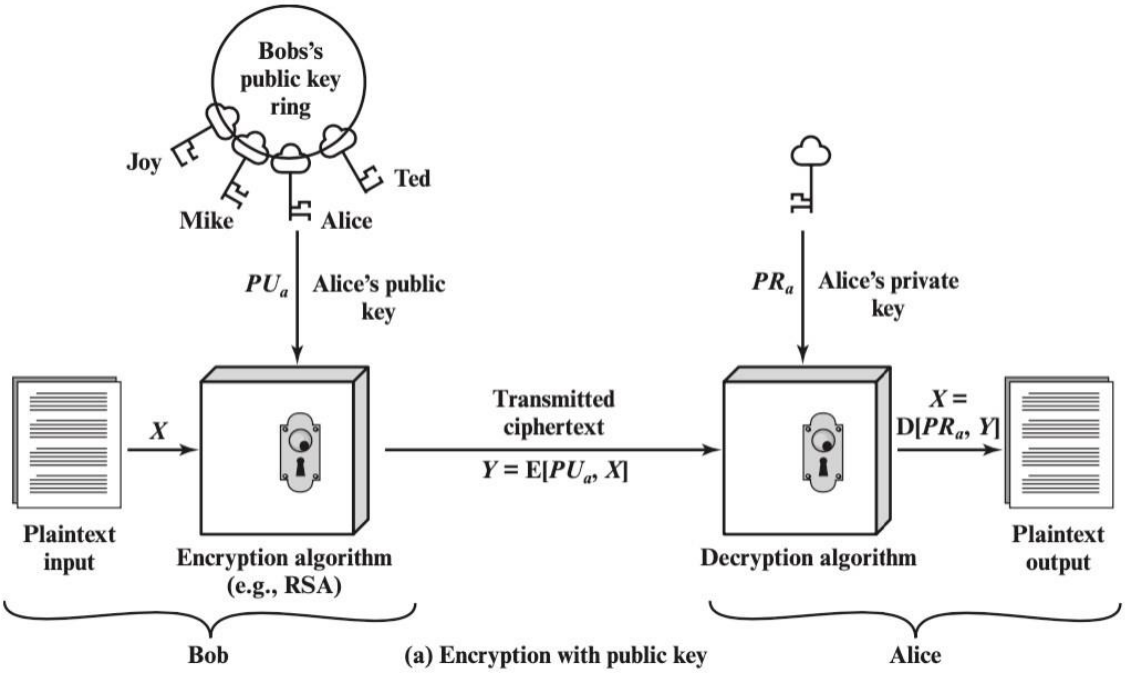
• keep the private key / secret key (SK) and distribute the public key (PK).

• Place PK in a public register or other accessible file.

Step3: Bob encrypts the message with Alice's PK.

Step4: Upon receiving the ciphertext (CT), Alice decrypt CT with SK.

Example:



Q.4. Which categories should be used to classify public key cryptography algorithms? (4 points)

Ans:
There are three categories used to classify Public key cryptography algorithms which are listed below:
1. **Encryption/Decryption:** It provides secrecy to the key.
2. **Digital signatures:** It provides authentication.
3. **Key exchange:** It consists of session keys.
• Some algorithms are suitable for all uses while the others are specific to one.
• Either of the two related keys can be used for encryption, with the other used for decryption.

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Diffie–Hellman	No	No	Yes
DSS	No	Yes	No
Elliptic curve	Yes	Yes	Yes

Q.5. Write RSA encryption and decryption algorithms. Suppose the public key {e, n}, and private key {d, n} are given. (4 points)
Ans:
RSA Encryption Algorithm:
Given a message M and a public key {e, n}, the RSA encryption algorithm works as follows:
1. Create an integer m from the message M such that $0 \leq m < n$.

2. Make the ciphertext c as c ≡ m^e (mod n).
3. Return the ciphertext c.

RSA Decryption Algorithm:
Given a ciphertext c and a private key {d, n}, the RSA decryption algorithm works as follows:

1. Compute the plaintext p as p ≡ c^d (mod n).
2. Convert the plaintext p back into the original message M.
   • It should be noted that step 2 of the RSA decryption process is required since it is possible that the plaintext p and the original message M are different.
   - Any encoding or formatting strategy that was employed to change M into m in the RSA encryption algorithm can be used to change p back into M.

Overall, the RSA encryption and decryption algorithms are relatively simple to implement and provide a high level of security for data transmission. However, the security of the algorithm relies on the difficulty of factoring large numbers, so it's important to use sufficiently large key sizes to prevent attacks.

Q.6.What are the possible attacks exploiting RSA's properties? (4 points)

Ans:
1. **Mathematical attacks**: Several approaches, all equivalent in effort to factoring the product of two primes. The defense against mathematical attacks is to use a large key size.
2. **Timing attacks:** These depend on the running time of the decryption algorithm.
3. **Chosen ciphertext attacks:** This type of attacks exploits properties of the RSA algorithm by selecting blocks of data. These attacks can be thwarted by suitable padding of the plaintext, such as PKCS1 V1.5 in SSL.

Q.7. What is meant by message authentication? (4 points)

Ans: Message authentication is concerned with:

1. Protecting the integrity of a message
2. Validating identity of originator
3. non-repudiation of origin (dispute resolution)

Q.8. What are the 3 approaches to achieve message authentication? (4 points)

Ans: Message authentication is the process of verifying the integrity and authenticity of a message. The 3 approaches to achieve message authentication are:

1. Message Encryption:

Message encryption by itself also provides a measure of authentication.

A. If symmetric encryption is used, then:

1. Receiver knows sender that who must have created it.
2. Since only sender and receiver know the key used.
3. Known content cannot be altered.

B. If public-key encryption is used:

1. Encryption provides no confidence of sender
2. Since anyone potentially knows public key
3. So, need to recognize corrupted messages

C. However, if

1. Sender signs message using their private key
2. Then encrypts with recipients' public key
3. Have both secrecy and authentication

But at cost of two public-key uses on message

2. Message Authentication Codes (MACs):
A MAC is a cryptographic technique that uses a secret key to generate a tag that is appended to the message. The recipient can verify the integrity of the message by recalculating the tag using the same key and comparing it with the received tag.

3. Digital Signatures:
A digital signature is a cryptographic technique that uses a private key to generate a signature that is attached to the message. The recipient can verify the authenticity and integrity of the message by using the corresponding public key to verify the signature.

Q.9. What are the pros and cons of single-key cryptography? (4 points)

Ans:

- **Pros:**
  - Encryption is fast for large amounts of data
  - Provide the same level of security with a shorter encryption key
  - By now, it's unbreakable to quantum computing
- **Cons**
  - Key distribution assumes a secure channel
  - Does not protect sender from receiver forging a message & claiming it's sent by sender
  - It does not scale well for large networks. It requires a separate key for each pair of communicating parties, which can result in a large number of keys to manage and protect.

Q.10. List and explain the requirements for a secure hash function (4 points)
Ans:

1. can be applied to any sized message M
2. produces fixed-length output h
3. is easy to compute h=H(M) for any message M
4. given h is infeasible to find x s.t. H(x)=h
one-way property or preimage resistance
5. given x is infeasible to find x' s.t. H(x')=H(x)
weak collision resistance or second pre-image resistant
6. infeasible to find any pair of x,x' s.t. H(x')=H(x)
strong collision resistance

Q.11. What is the weakness of hash function? (4 points)
Ans:

- **Collison**
- **Length extension attack**
- **Man-in-the-middle attack.**

Q.12. How many solutions to guarantee the integrity of hash function? (4 points)
Ans:

• A message digest created using a secret symmetric key is known as a Message Authentication Code (MAC), because it can provide assurance that the message has not been modified.
• The sender can also generate a message digest and then encrypt the digest using the private key of an asymmetric key pair, forming a digital signature. The signature must then be decrypted by the receiver, before comparing it with a locally generated digest

Q.13. Does hashes provide integrity? (4 points)
Ans:
Scenario 1:

- **Scenario**
  - Mozilla publishes a new version of Firefox on some download servers
  - Alice downloads the program binary
  - How can she be sure that nobody tampered with the program?
- **Idea: use cryptographic hashes**
  - Mozilla hashes the program binary and publishes the hash on its website
  - Alice hashes the binary she downloaded and checks that it matches the hash on the website
  - If Alice downloaded a malicious program, the hash would not match (tampering detected!)
  - An attacker can't create a malicious program with the same hash (collision resistance)
- **Threat model: We assume the attacker cannot modify the hash on the website**
  - We have integrity, as long as we can communicate the hash securely

Scenario 2:

- **Scenario**
  - Alice and Bob want to communicate over an insecure channel
  - David might tamper with messages
- **Idea: Use cryptographic hashes**
  - Alice sends her message with a cryptographic hash over the channel
  - Bob receives the message and computes a hash on the message
  - Bob checks that the hash he computed matches the hash sent by Alice
- **Threat model: David can modify the message and the hash**
  - No integrity!

Q.14. What is the definition and properties of message authentication code? (4 points)

Ans:

- **Two parts:**
  - KeyGen() → K: Generate a key K
  - MAC(*K*, *M*) → T: Generate a tag T for the message M using key K

- Inputs: A secret key and an arbitrary-length message
- Output: A fixed-length tag on the message

- **Properties**
- **Correctness: Determinism**
  - **Note:** Some more complicated MAC schemes have an additional Verify(K, M, T) function that don't require determinism, but this is out of scope
- **Efficiency:** Computing a MAC should be efficient
- **Security:** existentially unforgeable under chosen plaintext attack

Q.15. What are the properties of HMAC? (4 points)
Ans:

- **HMAC is a hash function, so it has the properties of the underlying hash too**
  - It is collision resistant
  - Given HMAC(K, M) and K, an attacker can't learn M – one way
  - If the underlying hash is secure, HMAC doesn't reveal M, but it is still deterministic
- You can't verify a tag T if you don't have K
- This means that an attacker can't brute-force the message M without knowing K

Q.16. How many key produced by HMAC? What are those keys? How to generate those keys? (4 points)
Ans:

- **will produce two keys to increase security.**
- **Output H[(K+ ⊕ opad) || H[(K+ ⊕ ipad) || M]]**
- **If key is longer than the desired size, we can hash it first, but be careful with using keys that are too much smaller, they have to have enough randomness in them.**

Additional:

- **Use K to derive two different keys**
  - opad (outer pad) is the hard-coded byte 0x5c repeated until it's the same length as K+
  - ipad (inner pad) is the hard-coded byte 0x36 repeated until it's the same length as K+
  - As long as opad and ipad are different, you'll get two different keys
  - For paranoia, the designers chose two very different bit patterns, even though they theoretically need only differ in one bit

Q.17. What is authenticated encryption? Explain in detail two approaches in which authenticated encryption can be achieved. From the two approaches, which one is better? (4 points)
Ans:

- **Authenticated encryption (AE):** A scheme that simultaneously guarantees confidentiality and integrity (and authenticity, depending on your threat model) on a message
- **Two ways of achieving authenticated encryption:**
  - Combine schemes that provide confidentiality with schemes that provide integrity
  - Use a scheme that is designed to provide confidentiality and integrity
  - Method 1: Encrypt-then-MAC

  First compute Enc(K1, M)

Then MAC the ciphertext: MAC(K2, Enc(K1, M))

• Method 2: MAC-then-encrypt

First compute MAC(K2, M)

Then encrypt the message and the MAC together: Enc(k1, M || MAC(K2, M))

• Method 1 is better. Always use encrypt-then-MAC because it is more robust to mistakes.

Q.18. What are the characteristics of the output hash function (deterministic or non-deterministic)? Justify your answer. (4 points)

Ans: Deterministic.

• No randomness: Identical inputs yield identical outputs

• It is implemented in deterministic hardware or software.

Q.19. What is a digital signature and describe its properties? (4 points)

Ans:

• Digital Signature: The result of a cryptographic transformation of data that, when properly implemented, provides a mechanism for verifying origin authentication, data integrity, and signatory non-repudiation

• Three parts:

• KeyGen() → PK, SK: Generate a public/private keypair, where PK is the verify (public) key, and SK is the signing (secret) key

• Sign(SK, M) → sig: Sign the message M using the signing key SK to produce the signature sig

• Verify(PK, M, sig) → {0, 1}: Verify the signature sig on message M using the verify key PK and output 1 if valid and 0 if invalid

• Properties:

• Correctness: Verification should be successful for a signature generated over any message

Verify(PK, M, Sign(SK, M)) = 1 for all PK, SK ← KeyGen() and M

• Efficiency: Signing/verifying should be fast

• Security: Same as for MACs except that the attacker also receives PK

Q.20. Describe the steps of RSA digital signature algorithm? prove correctness of RSA digital signature. (4 points)

Ans: Steps of RSA digital signature algorithm:

Step1:
Generate a hash value, or message digest, mHash from the message M to be signed.

Step2:
Pad mHash with a constant value padding1 and pseudorandom value salt to form M'

Step3:
Generate hash value H from M'

Step4:
Generate a block DB consisting of a constant value padding 2 and salt

Step5:
Use the mask generating function MGF, which produces a randomized out-put from input H of the same length as DB

Step 6:
Create the encoded message (EM) block by padding H with the hexadecimal constant bc and the XOR of DB and output of MGF
Step 7:
Encrypt EM with RSA using the signer's private key

Correctness of RSA digital signature

Theorem: $sig^e \equiv H(M) \bmod N$

Proof:

$$sig^e = [H(M)^d]^e \bmod N = H(M)^{ed} \bmod N$$

$$= H(M)^{k\phi(n)+1} \bmod N$$

$$= [H(M)^{\phi(n)}]^k \cdot H(M) \bmod N$$

$$= H(M) \bmod N$$

Q.21. What method has been used to guarantee RSA Digital Signature? (4 points)
Ans:

- **Necessary hardness assumptions:**
  - **Factoring hardness assumption: Given n large, it is hard to find primes pq = n**
  - **Discrete logarithm hardness assumption: Given n large, hash, and hash^d mod n, it is hard to find d**
- **Salt also adds security**
  - **Even the same message and private key will get different signatures.**

Q.22. What are the issues with public key encryption? Because of the issue what method we used to encrypt large size data that will be transmitted. (4 points)

Ans:

- **Issues with public-key encryption**
  - Notice: We can only encrypt small messages because of the modulo operator
  - Notice: There is a lot of math, and computers are slow at math
  - Result: We don't use asymmetric for large messages
- **Hybrid encryption: Encrypt data under a randomly generated key *K* using symmetric encryption, and encrypt *K* using asymmetric encryption**
  - Enc_{Asym}(PK, K); Enc_{Sym}(K, large message)
  - Benefit: Now we can encrypt large amounts of data quickly using symmetric encryption, and we still have the security of asymmetric encryption

Q.23. Short answer Questions (12 points)

1. _____ encryption is a form of cryptosystem in which encryption and decryption are performed using a public key and a private key.
2. Asymmetric encryption transform plaintext to ciphertext using _____
3. Asymmetric cryptography transforms plaintext into signature using_____
4. Public-key cryptography is asymmetric, involving the use of two separate keys, in contrast to _____ encryption, which uses only one key.
5. _____ is an example of homomorphic encryption.
6. The _____ has a function that is not vulnerable to _____.

Answers:
1. Asymmetric
2. Public Key
3. Private Key
4. symmetric
5. RSA
6. SHA-3, length extension attack