Homework 1

Problem 1 - Cryptography

1. What are the three properties that a Cryptographic Hash Function needs to satisfy? Explain the meaning of these properties in math.
2. (True or False) In digital signatures, the hash value of the message is encrypted with a user's public key. Explanation is needed.
3. Review RSA and Diffe-Hellman algorithms, and then answer the question. Which of the following algorithms can be used for digital signature: RSA, Elliptic Curve, Diffie-Hellman? Explanation is NOT needed.
4. (True or False) In ECDSA, the private key is an unpredictably chosen number between 1 and the order of the group. The public key is derived from the private key by scalar multiplication of the base point a number of times equal to the value of the private key.

Answer:

1.
   - Preimage Resistant
     For hash function:
     $$y = h(x) \tag{1}$$
     Given the output y of n bits, the time complexity for finding the preimage x is $O(2^n)$, the time complexity is great so it's hard to decode from the output y.
   - Collision Resistant
     For hash function h, it is infeasible to find two distinct values x and x such that:
     $$h(x) = h(x') \tag{2}$$
   - Second Preimage Resistant
     For hash function:
     $$y = h(x) \tag{3}$$
     it's infeasible to find another message x' to let:
     $$h(x) = h(x') \tag{4}$$
2. **False**
   The hash value of the message isn't encrypted with a user's public key, it's the user's private key.
3. **RSA** and **Elliptic Curve**
4. **True**

Problem 2 - Birthday Paradox
Prove the theorem regarding the birthday paradox in Lecture 1, slide 16.

Answer:

Firstly, we recall the Birthday Paradox mentioned in class.

Theorem: Let $r_1, r_2, \dots r_n \in \{1, 2, \dots N\}$ be independent, identically distributed integers, then

$$\Pr\left[r_i = r_j \mid i \neq j\right] \geq 1/2 \text{ for } n = 1.2 \times \sqrt{N}.$$

Proof:

$$\begin{aligned} \Pr\left(r_i = r_j \mid i \neq j\right) &= 1 - \Pr\left(r_i \neq r_j \mid i \neq j\right) \\ &= 1 - \prod_{i=1}^{n-1} \left(1 - \frac{i}{N}\right) \end{aligned} \tag{5}$$

And by the Bernoulli's Inequality $e^x \geq 1 + x$, we can treat $-\frac{i}{N}$ as x, therefore, we have:

$$\begin{aligned} \Pr\left(r_i = r_j \mid i \neq j\right) &\geq 1 - \prod_{i=1}^{n-1} \exp\left(-\frac{i}{N}\right) \\ &= 1 - \exp\left(-\sum_{i=1}^{n-1} \frac{i}{N}\right) \\ &= 1 - \exp\left(-\frac{1}{N}\frac{n(n-1)}{2}\right) \\ &\approx 1 - \exp\left(-\frac{1}{N}\frac{n^2}{2}\right) \end{aligned} \tag{6}$$

As for $n = 1.2 \times \sqrt{N}$, we have:

$$Original = 1 - \exp\left(-0.72\right) \approx 1 - 0.487 \geq \frac{1}{2} \tag{7}$$

Therefore,

$$\Pr\left[r_i = r_j \mid i \neq j\right] \geq 1/2 \text{ for } n = 1.2 \times \sqrt{N}. \tag{8}$$

Problem 3 - Bitcoin Transaction Signing

Recall that in Bitcoin, a user needs to provide a signature in his current transaction in order to spend his UTXO. Whenever a node validates a transaction, it checks the signature on exactly what was signed and rejects the transaction if the signature is invalid. For each transaction signing method listed below, decide if an attacker can steal funds from an input address of a transaction submitted to the Bitcoin network. Explanation is needed.

1. The private key is used to sign the entire transaction.
2. The private key is used to sign the entire output of the transaction and nothing else.

Answer:

1. The attacker cannot steal funds in this scenario since the entire transaction is signed with the private and it's infeasible for attacker to tamper anything to steal the funds.
2. It's possible for the attacker to steal funds in this scenario, if I have multiple UTXOs with the same public address, and spent first with output signed only, then the attacker can duplicate and temper the transaction, such as the transaction indentity, etc.

   By doing so, the attacker can use the duplicated transaction without my approval, in order to steal my funds.

Problem 4 - Bitcoin Confirmations

1. What does it mean when we say in Bitcoin a transaction is unconfirmed until it has n confirmations?
2. Why is it risky for the seller to accept a Bitcoin transaction with 0 confirmation?

Answer:

1. According the Longest Chain Rule, we can not make sure which chain is going to be the longest chain without the chain is confirmed after $n-1$ blocks created after this block on the main chain.
2. It's risky because the seller is facing the issues of double spending, such as a Finney attack by a malicious miner:
   - Includes a transaction sending some coins to himself in his mined block.
   - Withholds the mined block, and instead send the same coins to a merchant for some goods or service.
   - Broadcasts the block when the merchant accepts the payment and irreversibly provides the service.

Problem 5 - Bitcoin Scalability

1. Assume each block is mined in 10 minutes, a block has size 1M bytes, and each transaction has an average size of 250 bytes. What is the transaction per second (TPS) Bitcoin network can handle?
2. Scalability of the blockchain is currently a concern. To solve this issue, we can think about increasing the block size or shortening the block generation interval. What are their limitations in terms of increasing TPS?

Answer:

1. The TPS can be calculated through:

$$\text{TPS} = \frac{\text{Block Size}}{\text{Transaction Size}} \tag{9}$$

Therefore,

$$\text{TPS} = \frac{10^6(bytes) \times 10(mins)}{250(bytes) \times 60(seconds)} \approx 6.7(\text{transaction per second}) \tag{10}$$

2. Increasing the block size might lead to the hard fork requires waiting for sufficient consensus. Also, it leads to a greater risk of catastrophic consensus failure and greater propagation time.

   To shorten the block generation interval without increasing the incidence of forks, it is necessary to simultaneously shorten the block propagation time. This, in turn, influences the scale of the number of nodes in a blockchain network and the Proof-of-work difficulty.

   The oppositions to increasing TPS, are the concern of fork probability and increasing risk and complexity, those are the limitations in terms of increasing TPS.