INFO2222 Assignment 1

Mingyuan Ba

March 16, 2024

1 Part1

1.1 Q1

1. Web server handling online sales for the computer hardware parts.
    - **Confidentiality**: In this scenario,this principle need to ensures that sensitive data is accessible only to those authorized users[1]. A possible solution to this is encrypting data transmission using protocols like SSL, which helps in securing user data.
    - **Integrity**: This principle ensures that information keeps accurate during transmission[1]. It can be implemented by using hash function[2] and digital signature to validate the integrity of the data transmitted.
    - **Availability**: This principle ensures that users have reliable access to the website and do operations when needed[1]. Techniques to achieve availability include setting up load balancer[3], and utilizing robust hardware resources, ensuring that users can do operations when needed.
2. ATM machines
    - **Confidentiality**: This principle ensures that only authorized users and systems can access transaction details and user data[1]. It can be implemented by encrypting data transmission to secure customer PINs and transaction data from unauthorized access.
    - **Integrity**: This principle ensures that transaction information remains accurate during its transmission and processing[1]. In this scenario,Secure Socket Layer (SSL) and Transport Layer Security (TLS) [4]protocols not only provide an encrypted connection from the ATM to the host, protecting the data from leaks, but also ensure the integrity of the data during transmission.
    - **Availability**: This principle guarantees that customers can access services like withdrawals anytime they need[1]. To ensure continuous operation, ATMs can be equipped with backup power supplies and multiple network connections to handle hardware failures or network issues efficiently.

1.2 Q2

Reasons:

1. To use one-time pad, both the sender and receiver need to have the same key.But the issue is that if the key is compromised during transmission, the security of the encryption can not be promised.

   Example:
   Suppose that James and Alex are in Sydney and Melbourne separately.If they send key via internet or e-mail,the key could be intercepted by a third party during transit.Once the key is compromised, the security of the communication can no longer be guaranteed.

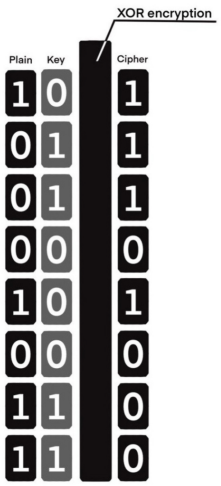
2. Since the key used in a one-time pad must have the same length of the message, managing and storing such long keys becomes hard. *Example*:
   If Rubin needs to send many large files to James, ensuring the keys are sent and managed correctly and securely would be a significant challenge for both parties.

3. The encryption and decryption process in a one-time pad is bit-by-bit, which can be computationally expensive and slow for large amounts of data.

   Example:
   Suppose that Alex wants to send a large 5GB file to Rubin using a one-time pad.This means for a 5GB file, they would need a key that is also 5GB in size, which equates to 40 billion bits. This process involves operating on every single bit of the file, resulting in an encrypted file of the same size.

Therefore, while the one-time pad provides perfect secrecy, it is not practical for most scenarios.

Stream ciphers are encryption algorithms that encrypt and decrypt messages by applying a cryptographic key and algorithm to each binary digit in a data stream, one bit at a time[5].



Pseudorandom generators are needed with stream ciphers to stretch a short key and generate a longer sequence of pseudorandom bits[6].

a graph here

1.3 Q3

The main difference between hash functions and MACs is that hash functions don't require a key for their operation. Hash functions produce a hash value for any input without a key. On the other hand, MACs use a secret key alongside data to produce a tag.It ensures the authenticity and integrity of the message for sender[6].

```
# By Hash
digest = Hash(M)

# By MACS
digest = MACS(M, key)
```

HMAC combines the properties of hash functions and MACS. HMAC uses a cryptographic hash function along with a secret key to generate a tag. This tag can be verified by the recipient using the same secret key and hash function. The use of a secret key makes HMAC more secure as it adds an additional protection against .

```
# By HMAC
K1 = preprocess_key(key)
K2 = preprocess_key(key)
inner_digest = Hash(K1 | M)
digest = Hash(K2 | inner_digest)
```

Using 2 keys in HMACS for internal and external hash isolates these processes.Even though the inner hash were compromised by third parties,they can not produce a digest to deceive the receiver without external key.

Compared to MACs, using HMAC for secure communication is like sending a locked box to someone, but with two keys. First, you use a secret code (K1) to lock the box inside another box. Then, you use a different secret code (K2) to lock the outer box. Even if someone figures out the first code (K1), they can't open the outer box without the second code (K2).

Besides,these two hash operations secure HMAC against a length extension where atttacker can add additional data to a message and generate a valid new hash value without knowing the original message content, only its hash[7].

The shortcomings of hash are the collision resistance and pre-image attacks[2].It is possible for two different messages to produce the same hash output, although the probability is extremely low.Pre-image attack involves attempting to reverse the hash function to determine the original input that generated a particular hash.

In HMAC or MAC, both the sender and the receiver have to know the key .This means that when sharing the key over the internet, it could be stolen by a third party.

1.4 Q4

In SHA-256, the number "256" refers to the bitsize of the output[8].

SHA-1 is considered "broken" for collision resistance [2], meaning that it is possible to find two different inputs that produce the same hash value. This property makes SHA-1 unsuitable for secure applications. In contrast, SHA-256 provides higher performance in collision resistance.

Differences between SHA-256 and MD5

- MD5 is produce output faster than SHA256. SHA256 is slower than MD5 due to its more complex algorithm and larger output size.
- MD5 is broken and insecure for cryptographic use because collision attacks can efficiently produce the same hash for different inputs[2]. On the contrary, SHA256 is more secure against such attacks.

Example

```
# Suppose s is a string with 1000000 characters

# Hash by MD5
md5_hash = hashlib.md5(s.encode()).hexdigest()

# Hash value: 174ac9a4f023a557a68ab0417355970e

# Hash by sha-256
sha256_hash = hashlib.sha256(s.encode()).hexdigest()

# Hash value: ec21d64624228af3ecd4bdaa8239e32ed943b01e26934cd5610fddb361
```

SHA-256 produce longer hash output and more secure against collision attacks.

References

[1] Lecture1,lecture slides. Canvas.

[2] 2-1-1.hash (1),lecture slides. Canvas.

[3] Alibaba Cloud. How to set up a web server with high availability using server load balancer. https://www.alibabacloud.com/blog/how-to-set-up-a-web-server-with-high-availability-using-server-load-balancer_598769, 2019. Accessed: 2024-03-16.

[4] Triton. Secure socket layer project. https://triton.com/project/secure-socket-layer/. Accessed: 2024-03-16.

[5] Okta. Stream cipher. https://www.okta.com/au/identity-101/stream-cipher/#:~:text=A%20stream%20cipher%20is%20an,both%20encrypts%20and%20decrypts%20messages., 2024. Accessed: 2024-03-16.

[6] 3-1.symmetric-crypto,lecture slides. Canvas.

[7] DeepRND. Length extension attack. https://deeprnd.medium.com/length-extension-attack-bff5b1ad2f70, 2019. Accessed: 2024-03-16.

[8] SSL Dragon. Sha-256 algorithm. 2024. Accessed: 16-Mar-2024.