INFO2222 Assignment 1

Mingyuan Ba

March 23, 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.

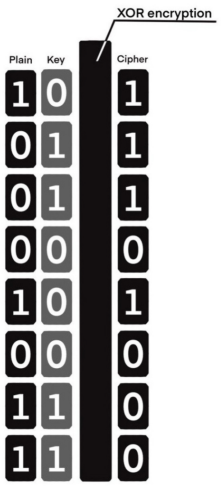
2. Since the key used in a one-time pad must have the same length of the message, managing and storing such long keys becomes hard. *Example*:
   If Rubin needs to send many large files to James, ensuring the keys are sent and managed correctly and securely would be a significant challenge for both parties.

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 produces output faster than SHA256. SHA256 is slower than MD5 because of its complex algorithm and larger output size[9].
- 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
```

As the example shown above demonstrates, SHA-256 produces a longer hash output and is more secure against collision attacks.

1.5 Q5

To achieve this, USYD can utilize a Key Derivation Function (KDF) to generate unique keys for students from the master key without the need to store each one. The KDF combines the master key and the SID to generate a unique key[10], which each student can use to access Canvas. This method ensures that even if a student's key is compromised, both the master key and the keys of other students remain secure.

2 Part2

2.1 Q1

1. Due to the independence of each attempt, the probability can be calculated by summing the probabilities of each time:

$$6 \cdot \frac{1}{10000} = \frac{6}{10000}$$

2. (a) **Identify Alice's Entry**: A locates the database entry corresponding to Alice, identified by her name and the hash of her password (name; H(pwd)).

   (b) **Understand the Hash Function**: A needs to figure out which hash function are used by the email server(e.g. sha-256).

   (c) **Use a Rainbow Table or Dictionary Attack**: Since hash functions are irreversible, A can use pre-computed tables, known as rainbow tables, which contain a large number of plaintext passwords and their corresponding hash values, to find a match for the hash value (name; H(pwd)) associated with Alice's password[11]. If the hash function used includes salt (a random value added to the password before hashing), this approach becomes much more challenging. A dictionary attack involves trying many passwords to see if they match.

   (d) **Reconstruct the Password**: Once A finds a match for the hash value using one of the methods above, they can reconstruct Alice's password with 100% confidence.

2.2 Q2

- **Advantages**

  1. *Convenience*: Biometrics such as fingerprints and facial recognition allow for quick identity verification through simple body movement,which typically completed in just a few second,while password verification usually takes longer time.Besides,users don't need to remember passwords or change them regularly to ensure security.
  2. *Security*:Compared with password verification, it is nearly impossible to replicate due to its uniqueness, unlike passwords which can be replicated.

- **Disadvantages**

  1. *Irreversibility*: Once a biometric attribute (e.g., fingerprint, iris pattern) is compromised, it cannot be easily changed or reset like a password.
  2. *Error Rates*: Biometric systems can mistakenly accept the wrong person or reject the right person. This mistake rate can change due to the technology, surroundings, or changes in a person's biometric features over time, leading to security risks or trouble accessing the system[12].

2.3 Q3

- **something you have**: The advantage is its physical security. Even if someone knows your password or personal details, they still can't access your account without the physical item. This method adds an additional layer of security, decreasing the risk of unauthorized access.
- **someone you know**: The benefit lies in convenience and not needing to remember passwords or carry physical items. It uses social networks for authorization, offering an easy way to regain access through known contacts.s.

2.4 Q4

1. **Q1**:

   During the TLS handshake phase, digital certificates and server authentication are used to prevent impersonation. In this process, the server presents its digital certificate, which is issued by a Certificate Authority (CA) and includes the server's public key. The client verifies the authenticity of the certificate by checking the digital signature against the CA's public key.

   If the certificate is deemed valid and trustworthy, the client can be assured that it is communicating with the actual server. This procedure relies on the fact that attackers cannot forge a valid certificate signed by a trusted CA, nor can they obtain the private key. Thus, without using a pre-master secret, the core of the entire process lies in the role of the CA and the verification of digital certificates

2. **Q2**: In TLS, each session generates a new random nonce as a nonce and uses it together with the key[13]. This means that even if an attacker replays a previously eavesdropped session, the message will be encrypted into a different ciphertext due to the use of a new random nonce and key, making it impossible to decrypt into the original message.

3. **Q3**:

   - *SQL injections*: TLS/SSL does not protect against SQL injections, which are a type of attack where an attacker inserts malicious SQL code into a query, allowing them to manipulate or retrieve sensitive data from a database[13].
   - *Client/Server side vulnerabilities*: TLS/SSL cannot protect against vulnerabilities in the client or server applications themselves[13]. If either the client or server has a vulnerability, an attacker can exploit it to gain unauthorized access or manipulate data, regardless of the TLS/SSL encryption in place.

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] Thomas Peyrin, Yu Sasaki, and Lei Wang. Generic related-key attacks for hmac. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, pages 580–597, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

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

[9] Shwetha R. Prasanna and B.S. Premananda. Performance analysis of md5 and sha-256 algorithms to maintain data integrity. In *2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, pages 246–250, 2021.

[10] Carlisle Adams, Guenther Kramer, Serge Mister, and Robert Zuccherato. On the security of key derivation functions. In *International Conference on Information Security*, pages 134–145. Springer, 2004.

[11] 3-1.symmetric-crypto,lecture slides. Canvas.

[12] Lawrence O'Gorman. Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12):2021–2040, 2003.

[13] 5-2.security protocol,lecture slides. Canvas.