INFO2222 H1 - Homework 1

Computing 2 Usability and Security (University of Sydney)

INFO2222 H1

Q1)

Webserver

Confidentiality:

A web server handling online sales information would have confidentiality concerns with regard to the privacy of consumer and transaction information and how to prevent this sensitive data from being accessed by unauthorized personal. This can be implemented through SSL protocols and firewalls, encrypting communication between the server and the client, and ensuring sensitive information, like credit card details, is transferred securely [1].

Integrity:

Integrity could be implemented in this web server by maintaining the consistency of customer data. This is seen in ensuring the order details are valid and consistent, unmanipulated by unauthorised people. This can be implemented by hashing the data about the items in the client's cart whilst browsing, then doing so again during checkout to determine if they have been altered.

Availability:

Proper availability in a web server would entail ensuring that the server is always running so customers have access to see hardware parts to buy. This would be done by implementing backup and disaster plans to overcome potential risks that would take down the server. For example, making sure the server can handle a large amount of people viewing its website it at once by utilising load balancing to distribute traffic across different servers [2].

ATM

Confidentiality:

ATMs would need to maintain confidentiality when transmitting sensitive information like PIN numbers and account information. They need to protect this data from unauthorised access attempts. This can be done through encryption methods and secure authorization protocols.

Integrity:

Transactions done through an ATM network should be accurately and securely recorded to ensure that the correct amounts are credited or debited to the correct accounts. This can be implemented through data validation checks, audit logs, and encryption of data to prevent information from being tampered.

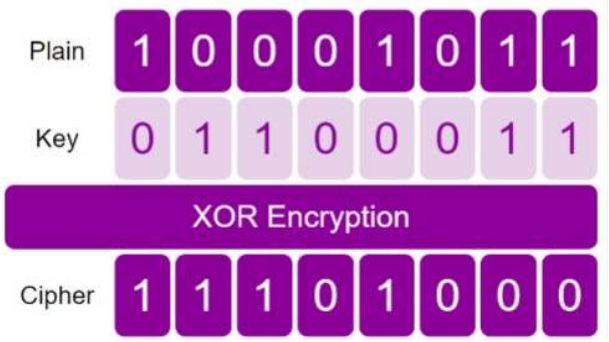
Availability:

The ATM network should be available for use by authorized users at all times and should be able to process transaction requests quickly and efficiently. This can be implemented through potential disaster measures to ensure uninterrupted availability. For example, monitoring and detection systems to detect and prevent network disruptions, and regular maintenance and servicing of the network infrastructure.

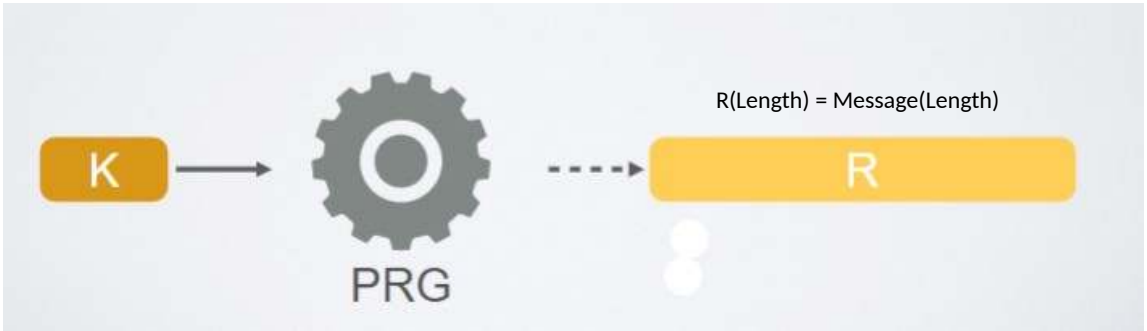
Q2)

One time pad encryption is not used in practice as the key system is inefficient and they are more vulnerable to attacks than other forms of encryption. The key length has to be as long as the message, can only be used once and must be independently distributed to the receiver of the message [3]. As such, OTP is an impractical encryption technique when handling frequent lengthy communications like for an internet communication company such as Snapchat. In this instance, using OTP would require a large key distribution system to create and distribute the keys and be able to handle the possibly extremely lengthy keys from large messages sent. Furthermore, if information is known about the plain text of a particular message of an OTP encrypted message it is vulnerable to being broken. As there are much more secure and efficient encryption techniques that not only require less but also shorter keys, OTP is not used in practice.

Stream ciphers are a form of symmetric key cipher that takes the plaintext digits of a message and a keystream of same length and performs an XOR operation one at a time on each of the coinciding bits from the message and keystream [4]. These new bits form the digits of the new ciphertext.



Thus, we need pseudorandom generator to expand a key to the length of the message so each bit from the message has a corresponding bit in the keystream [5], As keys are not always as long as the message.



Q3)

The difference between a MAC and a hash function is that anyone can take a message and use a public hash algorithm to compute the hash value, whilst in a MAC , the message is first combined with a secret key before it is inputted into a hash function.

In Hashing:

- H(Message) = digest

In MAC:

- H(Message combined with Key) = digest

In principle, a MAC can better ensure integrity than just hashing messages as both the sender and receiver of the message must have access to the secret key to produce the valid hash digest [6].

HMAC's look more secure as it derives two keys from the shared secret key and performs two hash operations as such:

In HMAC:

- $H(K_2 \mid H(K_1 \mid M))$ = digest

These two hash operations make HMAC secure against a length extension attack. This is because the result of the inner hash produces a fixed length string. An attacker can only manipulate the length of the input into the inner hash [7].

The shortcomings of hash are their vulnerability to collision and pre-image attacks. Collision attacks occur when two different inputs produce the same hash output. A preimage attack is a result of an attacker finding an input that hashes to a specific output. In these cases, the hash function does not provide message integrity.

HMACs do not have these shortcomings' part are instead vulnerable to key compromised attacks. If an attacker finds the secret key, they can use it to generate HMAC codes and can break a messages integrity.

Q4)

The 256 in SHA-256 refers to the bit length of the hash value, being 256 bits long. SHA-256 is better than previous versions of SHA as it is more secure. It uses a larger message digest and more rounds of computation. SHA – 1 produces a 160-bit length digest compared to SHA – 256's 256-bit length digest. This extra length makes SHA-256 more resistant to collision and brute force attacks [8].

The main differences in performance for SHA-256 and MD5 is as follows:

- SHA-256 produces a more secure digest than MD5. MD5 generates a MD5 generates a 128-bit long hash value and thus is more vulnerable to collision attacks.
- SHA-256 is slower than MD5, as it has a more complex algorithm. SHA-256 uses 64 rounds of hashing compared to MD5's 4, increasing its complexity. Further, SHA-256's larger block size operations of 512-bit blocks compared to MD5's 128-bit blocks, requires more processing power and thus more time to generate [9].

For example, consider the plain text message “Hello world”.

- MD5 hash value:
  5eb63bbbe01eeed093cb22bb8f5acdc3
- SHA-256 hash value:
  b94d27b9934d3e08a52e52d7da7dabfac484efe37a5380ee9088f7ace2efcde9

[10]

SHA-256’s digest is longer and more resistant to attacks than MD5’s. The processing time difference in this instance was negligible.

Q5)

To produce a unique key for student without having to store them, USYD can use a Key derivation function to create unique keys for each student using the short secret and the student’s unique SID . The KDF would apply a one-way function on the msk and SID to create a unique key for that student [11]. When a student logs into canvas and initiates a communication session, canvas uses the KDF to generate a unique key for that student in that session. In turn, USYD can generate and distribute unique keys without needing to store all of them.

Q6)

To convince the students the public key is the correct one, the lecturer must first convince the students they are a lecturer. The lecturer can do this by sending the students their teacher’s certificate and a digital signature of the certificate signed with the VC’s private key. The students can then use the VC’s public key to verify the certificates and lecturers’ authenticity. This is because the VC is a trusted authority to the students. Then the lecturer must prove the integrity of the original email containing the public key. They can do this by sending the students a digital signature of the original message signed with their private key. The students can use the public key in the original message to verify the digital signature and thus prove that the public key is from the correct one for the lecturer.

Q7)

To upload the 5 Gigabytes video quickly and securely to Google Cloud I would use a combination of symmetric and public key encryption. The mechanism is a hybrid encryption scheme and goes as follows:

1. First create a random symmetric key (K_1) to encrypt the video file.
2. Encrypt the video with K_1 with a symmetric encryption algorithm (AES)
3. Encrypt K_1 with googles public key pk_G using a public key encryption algorithm (RSA)
4. Send the encrypted video and the encrypted K_1 to Google Cloud
5. Once uploaded, Google Cloud can use its private key to decrypt and find K_1
6. Google Cloud can then use K_1 to decrypt the symmetrically encrypted video file.
7. Google Cloud can then view the video file.

By using hybrid encryption, the large video file is encrypted quickly with symmetric encryption. Further, by using public key encryption on the symmetric key google doesn't have to share a secret key with its customer [12].

Referencing

[1]"8 Ways to Protect E-commerce Customer Data," *SEMrush Blog*. https://www.semrush.com/blog/8-ways-to-protect-ecommerce-customer-data/

[2]K. Yasar, "What is load balancing?," *SearchNetworking*. https://www.techtarget.com/searchnetworking/definition/load-balancing (accessed Mar. 31, 2023).

[3]R. Wright, "One-Time Pad - an overview | ScienceDirect Topics," *www.sciencedirect.com*, 2003. https://www.sciencedirect.com/topics/mathematics/one-time-pad#:~:text=One%2Dtime%20pads%20are%20impractical (accessed Mar. 31, 2023).

[4]Okta, "Stream Cipher 101: Definition, Usage & Comparisons - Okta AU & NZ," *www.okta.com*, Feb. 14, 2023. https://www.okta.com/au/identity-101/stream-cipher/ (accessed Mar. 31, 2023).

[5]D. Khurana, "Stream Cipher Examples, Block Cipher Introduction," *University of Illinois, Urbana Champaign*, Sep. 01, 2020. https://courses.grainger.illinois.edu/cs498ac3/fa2020/Files/Lecture_3_Scribe.pdf (accessed Mar. 31, 2023).

[6]Computerphile, "Securing Stream Ciphers (HMAC)," *www.youtube.com*, Aug. 24, 2017. https://www.youtube.com/watch?v=wlSG3pEiQdc&t=300s&ab_channel=Computerphile (accessed Mar. 31, 2023).

[7]S. Touset, "length extension - How does the secret key in an HMAC prevent modification of the HMAC?," *Cryptography Stack Exchange*, Jan. 02, 2014. https://crypto.stackexchange.com/questions/12680/how-does-the-secret-key-in-an-hmac-prevent-modification-of-the-hmac (accessed Mar. 31, 2023).

[8]KeyCDN, "SHA1 vs SHA256 - KeyCDN Support," *KeyCDN*, Feb. 22, 2023. https://www.keycdn.com/support/sha1-vs-sha256#:~:text=If%20you%20are%20using%20SHA1%20for%20password%20hashing%2C%20you%20should (accessed Mar. 31, 2023).

[9]A. Stec, "MD5 vs. SHA Algorithms," *Baeldung*, Nov. 06, 2022. https://www.baeldung.com/cs/md5-vs-sha-algorithms#:~:text=First%20of%20all%2C%20MD5%20produces,in%20terms%20of%20collision%20resistance. (accessed Mar. 31, 2023).

[10]Browserling, “Generate All Hashes - MD5, SHA1, SHA3, CRC32 - Online - Browserling Web Developer Tools,” *www.browserling.com*. https://www.browserling.com/tools/all-hashes (accessed Mar. 31, 2023).
[11]J. Lake, “What is a key derivation function (KDF) and how do they work?,” *Comparitech*, Aug. 14, 2022. https://www.comparitech.com/blog/information-security/key-derivation-function-kdf/ (accessed Mar. 31, 2023).
[12]Tink Cryptographic library, “Hybrid encryption | Tink,” *Google Developers*. https://developers.google.com/tink/hybrid#:~:text=Hybrid%20Encryption%20combines%20the%20efficiency (accessed Mar. 31, 2023).