1. Fundamentals Research Questions:

What are cryptographic hash functions?

A cryptographic hash function is an algorithm that turns data input into a sequence of bits of fixed length. In cryptography these functions are widely used because they allow the integrity of a message to be verified without knowing the contents of the input data (Stinson & Paterson, 2018).

What are the main properties of cryptographic hash functions?

Doing some research, one can conclude that there are three main properties for such functions (Katz & Lindell, 2020):

1. Preimage resistance: given a hash **h**, it should be computationally impossible to find a message **m** were

$$hash(m) = h$$

As seen in class, one of the concepts that cybersecurity tries to deal with is data confidentiality which, after being dealt properly, prevents external users to reconstruct the original message **m** from the knowledge of **h** (Koblitz & Menezes, 2011).

2. Second preimage resistance: given an input m_1, it should be computationally impossible to find another input message m_2 such that

$$hash(m_1) = hash(m_2)$$

Hence, this concept deals with data intengrity by not letting external users to modify an original message and keep undetected (Rogaway & Shrimpton, 2004).

3. Collision resistance: it should be computationally impossible to find two different messages m_1 and m_2 for which

$$hash(m_1) = hash(m_2)$$

This property prevents not only data integrity but also data authenticity, considering attacks where an attacker could replace one valid input with another that generates the same hash (Damgård, 1989).

Note: The difference between property 2 and property 3, relies in the fact that collision resistance does not consider a given or fixed message, which implies a much wider view of avoiding hash collisions.

How do these properties contribute to security?

By ensuring that the data cannot be obtained using brute force, changed, or duplicated without detection, considering and protecting the integrity, authenticity, and confidentiality of the original information/data that can be used for sensitive processes such as: billing information, password storage, signatures and so on (Katz & Lindell, 2020).

2. Common Hash Functions Research Questions:

What are the most used hash functions?

The most used hash functions in terms of cybersecurity and cryptography include MD5, SHA-1, SHA-2 and SHA-3 (National Institute of Standards and Technology [NIST], 2015):

- MD5 was created by Ronald Rivest in 1991, and it was very popular because of its simplicity and quickness.
- During the last few years, the SHA functions became dominat in applications that needed advance security. SHA-1 was developed by the NSA, it was published in 1993, and it was vastly implemented in security systems until weaknesses in its collision system were discovered. Nowadays, the group or **family** SHA-2 (which includes, SHA-224, SHA-256, SHA-384 and SHA-512) is used instead because of its reliability.
- SHA-3 was developed in 2015, and it was adopted as a replacement for the SHA-2 making significant improvements on the collision system and offering different variations in security.

What are the differences between MD5, SHA-1, and SHA-2 family?

Their difference can be focused and explained around the length of the hashes that generate different complexity between these algorithms:

- **MD5:** Generates hashes of 128 bits. The simplicity and velocity of this function made it very popular to verify the integrity of files but nowadays is considered unsafe because of its different vulnerabilities on the collision systems (as discovered in 2004) which implied that one can generate different messages using the same hash MD5, compromising the integrity of the data (Wang & Yu, 2005).
- **SHA-1:** Produces 160 bits hashes and was adopted as a standard successor for the MD5. Additionally, it offers a better security management in comparison to the MD5, however, as shown by NIST and Google whom demonstrate how to make a successful attack to SHA-1 on 2017, it can be susceptible to collision attacks, stating that even though this function is more secure that MD5 it still does not deal with different risks in an environment that needs high standards of security (NIST, 2012).
- **SHA-2:** This family of functions have different versions of SHA-2: SHA-256 and SHA-512, which (respectively) can produce 256 and 512 bits. Nowadays, these

functions demonstrated that they can resist different collision attacks, hence being very reliable (NIST, 2012).

Why have some hash functions been deprecated?

The security of a hash function depends on its capacity to resist collision attacks. The deprecation of MD5 and SHA-1 was due to new techniques of cryptanalysis that demonstrated that they do not meet the modern security that most of the applications needed and nowadays have. Over time, vulnerabilities that involve collision attacks have become much easier to exploit thanks to the advancement of computing power, which is why different organizations recommend not using these functions (Wang & Yu, 2005).

What makes SHA-3 different from previous hash functions?

SHA-3 was finalized in 2015 and is different compared to SHA-2 and the other ones on the structural design. Instead of using the compression structure that is used on the MD5, SHA-1, SHA-2, SHA-3 use a different approach based on sponges, this was developed as part of the algorithm Keccak. This sponge structure allows SHA-3 to be more adaptable and resistant to collision attacks and pre-image (Bertoni et al., 2011).

One of the most important advantages that SHA-3 have is the capacity to absorb the inputs of variable longitude and squeeze outputs that have variable longitude, giving flexibility by different applications. Also, SHA-3 was not designed to replace SHA-2 directly but to act as a secure alternative if one day the SHA-2 becomes insecure. Nowadays SHA-3 is used in applications that require high standards of security and resistance against complex attacks (NIST, 2015).

3. Applications Research Questions:

How are hash functions used in digital signatures?

They create a unique hash of the message, which is then encrypted with the sender’s private key. The encrypted hash is the digital signature, which ensures that the message has not been altered and verifies the sender’s authenticity (Menezes, van Oorschot, & Vanstone, 1996).

What role do hash functions play in password storage?

They store passwords by hashing them before storage. When a user enters a password, it is hashed and compared to the stored hash, preventing the storage of plain-text passwords and protecting against unauthorized access (Katz & Lindell, 2020).

How are hash functions used in blockchain technology?

They ensure data integrity across the blockchain network by hashing each block’s data. Each block’s hash is included in the subsequent block, creating a chain that links all blocks, making it (almost) impossible to alter data without detection (Nakamoto, 2008).

What is an HMAC, and why is it important?

A Hash-Based Message Authentication Code (HMAC) is a mechanism for verifying message integrity and authenticity. It uses a cryptographic hash function and a secret key, ensuring the message's integrity and authenticity even if transmitted through an untrusted channel (Krawczyk, Bellare, & Canetti, 1997).

References:

Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (2011). The Keccak SHA-3 Submission. *Submission to NIST*. Retrieved from https://keccak.team

Damgård, I. (1989). A Design Principle for Hash Functions. *Advances in Cryptology – CRYPTO '89*, 416–427. Retrieved from SpringerLink

Katz, J., & Lindell, Y. (2020). *Introduction to Modern Cryptography* (3rd ed.). CRC Press. Available at Google Books

Koblitz, N., & Menezes, A. (2011). The Random Oracle Model: A Twenty-Year Retrospective. *Designs, Codes, and Cryptography*, 62(1), 7–26. Retrieved from SpringerLink

Krawczyk, H., Bellare, M., & Canetti, R. (1997). HMAC: Keyed-Hashing for Message Authentication. *Internet Engineering Task Force*, RFC 2104. Retrieved from https://datatracker.ietf.org/doc/html/rfc2104

Menezes, A., van Oorschot, P., & Vanstone, S. (1996). Handbook of Applied Cryptography. CRC Press. Retrieved from https://cacr.uwaterloo.ca/hac/

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf

National Institute of Standards and Technology (NIST). (2012). *Recommendation for Applications Using Approved Hash Algorithms*. Special Publication 800-107 Revision 1. Retrieved from NIST

National Institute of Standards and Technology (NIST). (2015). *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. Retrieved from NIST

Rogaway, P., & Shrimpton, T. (2004). Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. *Fast Software Encryption*, 371–388. Retrieved from SpringerLink

Stinson, D. R., & Paterson, M. B. (2018). *Cryptography: Theory and Practice* (4th ed.). CRC Press.

Stevens, M., Bursztein, E., Karpman, P., Albertini, A., & Markov, Y. (2017). The First Collision for Full SHA-1. *Crypto Journal*.

Wang, X., & Yu, H. (2005). "How to Break MD5 and Other Hash Functions." *Advances in Cryptology – EUROCRYPT 2005*.