Cryptographic Hash Functions

Juan Antonio Romero 00212936

Diego Reinoso 00214020

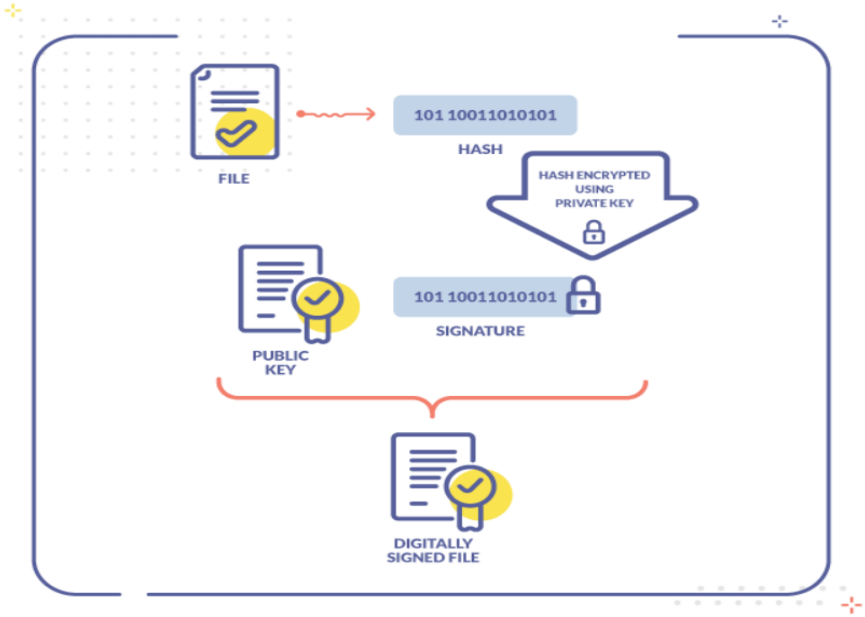
Cryptographic hash functions are mathematical algorithms used to convert data into a fixed-length string, known as the hash value or digest. These functions are designed to ensure data integrity and security in various applications such as digital signatures, password protection, and blockchain technology. A secure hash function exhibits three key properties: pre-image resistance, second pre-image resistance, and collision resistance. Pre-image resistance means that given a hash value, it should be computationally infeasible to find an input that produces that specific hash. This property is essential because it ensures the function is "one-way," making it impossible to reverse-engineer the original data from its hash output. The second pre-image resistance, often called weak collision resistance, prevents attackers from finding a second distinct input that yields the same hash as a given input. This ensures the uniqueness of hash outputs for different inputs, safeguarding against tampering or unauthorized modifications. Finally**,** collision resistance guarantees that it is difficult to find any two different inputs that result in the same hash output. This property is crucial for maintaining the integrity of systems relying on cryptographic hash functions, as a collision could potentially lead to a security breach where two different sets of data are indistinguishable by their hash values. Collectively, these properties protect against malicious activities like data manipulation, ensuring that cryptographic hashes remain a cornerstone of secure digital systems.

Hash functions are algorithms that convert any input, such as text or data, into a fixed-length string known as a hash value. Common examples include MD5, SHA-1, SHA-2, and SHA-3, each designed for tasks like verifying data integrity and securing passwords. However, MD5 and SHA-1, once popular, have proven vulnerable to "collisions"—situations where different inputs yield the same hash value. This vulnerability compromises their effectiveness in cryptographic tasks, as “they are no longer recommended for secure cryptographic practices (EITCA, 2023).” Because collisions can allow attackers to forge data, these older functions are now deprecated for security-sensitive uses, replaced by stronger algorithms like SHA-2 and SHA-3 for enhanced reliability and security.

In contrast, SHA-2, which includes SHA-256 and SHA-512, is much more secure and widely used today. SHA-2 produces longer hash values (256 or 512 bits), making it harder for attackers to find a match or manipulate data undetected. SHA-2 is the standard for many security protocols, such as SSL/TLS used in websites to encrypt data and ensure secure connections. SHA-3, the newest member of the Secure Hash Algorithm family, was designed to improve on the shortcomings of previous functions. While SHA-2 is still widely used, SHA-3 offers enhanced security and flexibility by using a different cryptographic method. It also allows for variable-length hashes, which can be useful for specific security needs. This makes SHA-3 an attractive option for applications requiring high security, and it is gradually being adopted in newer systems. Each of these hash functions plays a crucial role in protecting sensitive information, but the choice of which one to use depends on the level of security needed and the potential threats involved.

Hash functions are widely used in several important applications, such as digital signatures, password storage, blockchain technology, and HMACs, playing a crucial role in ensuring security and integrity in various systems.

In digital signatures, hash functions help in verifying the integrity of a message. When a document or message is signed, a hash of the message is created and encrypted with the sender's private key. The recipient can then use the sender's public key to decrypt the signature, obtaining the original hash, and comparing it with a hash of the received message. If the hashes match, the message has not been altered, and its authenticity is verified. This process is critical for verifying identities and ensuring data integrity, especially in digital communication and transactions. As said in this article, hashing and digital signatures help verify that a message hasn't been altered during transit by comparing the computed hash with the transmitted one. “As a result, the use of hashing and digital signature in blockchain could help recipients in recomputing the output of a hash function with the same hash function. The comparison of the message digest with the transmitted digest could help in verifying that the message didn’t go through unwanted modifications in transit (Blockchain 101, n.d.).” What is said in this article is the perfect example of how hashing and digital signatures can be used in blockchain to ensure data integrity, allowing recipients to verify that the transmitted message has not been altered by comparing its computed hash with the original



digest.

.

In the image above we have an example of how the file has a hash, this hash is encrypted with the private key. Then easily the receiver can decode this using the senders public key and eventually getting the digital signature and its file contents validating that all the info is correctly and safe as well because signatures are a form of trust and authenticity so its very important that they stay as confidential and as secure as possible.

Hash functions also play a significant role in password storage by converting passwords into a fixed-length hash, making it difficult to reverse-engineer the original password.” Hash functions play a crucial role in password security for secure web services. These services never store your actual password but instead store a hashed version of it. Hash functions are algorithms that convert input data (like a password) into a fixed-size string of characters (Meta Compliance, n.d.).” When users log in, the system hashes their entered password and compares it to the stored hash. Since hash functions are one-way functions, the actual password is never stored, providing security even if the database is compromised. A real-life example of where hash functions are used is in online banking systems. When you create an account or log into your bank's website, your password is hashed before it is stored in their database. Even if hackers breach the database, they will only access the hashed version of your password, not the actual password itself, making it much harder for them to compromise your account.

In blockchain technology, hash functions are vital for ensuring the immutability and security of transactions. Every block in a blockchain contains a hash of the previous block, which creates a chain of blocks that is resistant to tampering. If even a small change is made to the data in a block, the hash would change, making it obvious that the data has been altered. A perfect real-life example is crypto currency, let’s say you make a transaction with Bitcoin, this transaction will be hashed and stored into a block. Each block creates a blockchain that has a immutable chain making it secure.

"Finally, HMAC (Hash-based Message Authentication Code) combines a hash function with a secret key to ensure both data integrity and authentication. It is commonly used in secure communications like SSL/TLS protocols to verify the authenticity of messages and ensure that they haven't been tampered with during transmission (Fiveable, n.d.)."When making requests to a cloud service like AWS, an HMAC is generated using a secret access key and the request details. This HMAC is included in the request header. When the cloud service receives the request, it recomputes the HMAC using the same secret key and verifies if it matches the sent HMAC. This process ensures that the request was not altered during transmission and is indeed from an authorized user. We also need to wonder where we can use HMAC because it’s a nicely way to stay secure now in days.” Nearly every company has sensitive information. If you take in payments of any sort, for example, you likely have credit card data at your fingertips. And if you have employees, you have Social Security numbers that could be stolen (Otka, n.d.).” As said in this article now in days almost every company needs this being because of the employee’s private info or the customers so having HMAC as a tool can provide this type of secrecy and privacy between two parties.

In conclusion cryptographic hash functions play a crucial role in maintaining the security and integrity of digital systems. By converting data into unique, fixed-length hashes, they prevent unauthorized access and tampering, ensuring that sensitive information, like passwords and transactions, remain safe in various applications such as digital signatures and blockchain technology.

APA CITATIONS

EITCA Academy. (2023). *How are hash functions used in digital signatures and data integrity checks?* EITC/IS/ACC Advanced Classical Cryptography. Retrieved from https://eitca.org/cybersecurity/eitc-is-acc-advanced-classical-cryptography/hash-functions/introduction-to-hash-functions/examination-review-introduction-to-hash-functions/how-are-hash-functions-used-in-digital-signatures-and-data-integrity-checks/

Uanataca. (n.d.). *Hash and electronic signature*. Retrieved from https://web.uanataca.com/en/blog/technology/hash-and-electronic-signature

101 Blockchains. (n.d.). *Hashing and digital signature in blockchain*. 101blockchains.com. Retrieved November 12, 2024, from https://101blockchains.com/hashing-and-digital-signature-in-blockchain/

Okta. (n.d.). *HMAC: Hash-based Message Authentication Code*. Okta. Retrieved November 12, 2024, from https://www.okta.com/identity-101/hmac/

Fiveable. (n.d.). *Cryptographic protocols and secure communication study guide*. Fiveable. Retrieved November 12, 2024, from https://library.fiveable.me/quantum-cryptography/unit-3/cryptographic-protocols-secure-communication/study-guide/XoZCInsD9AZrGDS5.

DebugPointer. (2021, October 1). *SHA1 vs SHA2 vs SHA3: Differences and comparison*. DebugPointer. Retrieved November 12, 2024, from https://debugpointer.com/security/sha1-vs-sha2-vs-sha3

MetaCompliance. (n.d.). *Hash functions and information security*. MetaCompliance. Retrieved November 12, 2024, from https://www.metacompliance.com/blog/security-awareness-training/hash-functions-and-information-security#:~:text=Hash%20functions%20play%20a%20crucial,fixed-size%20string%20of%20characters.

.