CRYPTOGRAPHIC ALGORITHMS: SHA-3

Rouaa Naim
Hams Gelban
Sara Said
Vasiliki Gerokosta
Noora Al-Emadi

AGENDA

1. Introduction to the “Secure Hashing Algorithms” family
2. How does SHA-3 work?
3. Properties
4. Demo

THE “SHA” FAMILY

The “Secure Hashing Algorithm” family is a group of cryptographic hash functions developed by the National Security Agency (NSA) of the United States

THE “SHA” FAMILY

The “Secure Hashing Algorithm” family is a group of cryptographic hash functions developed by the National Security Agency (NSA) of the United States

SHA algorithms aim to ensure data integrity, authenticity, and non-repudiation by generating fixed-size hash values from arbitrary sized input data.

THE “SHA” FAMILY

The “Secure Hashing Algorithm” family is a group of cryptographic hash functions developed by the National Security Agency (NSA) of the United States

SHA algorithms aim to ensure data integrity, authenticity, and non-repudiation by generating fixed-size hash values from arbitrary sized input data.

SHA-3, or Secure Hash Algorithm 3, is the latest addition to the SHA family, offering better security and performance, compared to previous algorithms

SHA-1 VS. SHA-2 VS. SHA-3

	SHA1	SHA2 (224, 256, 384, & 512)	SHA3 (224, 256, 384, & 512)
Launch Year	1995	2001	2008
Block Size	512 bits	• 512 bits • 1024 bits	• 1152 bits • 1088 bits • 832 bits • 576 bits
Output (Hash Value or Message Digest)	160 bits / 20 bytes	• 256 bits / 32 bytes • 512 bits / 64 bytes	• 224 bits / 28 bytes • 256 bits / 32 bytes • 384 bits / 48 bytes • 512 bits / 64 bytes
Construction System	Merkle-Damgård	Merkle-Damgård	Sponge (Keccak)
Possibility of Collision	Possible - Google found proof of collision in 2017	No proof of collision has been found yet.	Susceptible to collision in squeeze attack.
Weakness	• Has only one use case - password storage. • Susceptible to collision Short key length	• SHA 256 is slower than its previous versions. • Softwares and browsers must be updated to implement SHA2.	• Susceptible to collision
Is it Still in Use?	No	Yes	Yes
Utility or Applications	Transport Layer Security (TLS)/ Secure Sockets Layer (SSL) Certificate Verifying the Integrity of a file	Security application protocols Cryptographic transactions Digital certificates	Can replace SHA2, where necessary.

SHA-2 OR SHA-3?

In terms of security, both SHA-2 and SHA-3 are highly secure and resistant to various cryptographic attacks. However, SHA-3 is less vulnerable to certain type of attacks (collision attacks, length extension attacks), due its design and different underlying principles.

HOW DOES SHA-3 WORK?

SPONGE CONSTRUCTION

"Padding Rule: how to add extra stuff to a string to make it the right size. It ensures that the string is the correct length for "f" to work on.

a bit string

the bit length of the output string

SPONGE[f, pad, r](N, d)

"The Function: takes strings of a certain length (called "b") and turns them into strings of the same length.

the rate: a number that's smaller than "b." The difference between "b" and "r" is called "c." So, "r + c = b."

"c" is the capacity, indicating the portion of the state that is not used for input and plays a role in the security of the construction.

THE STEPS

We have an empty "container" called "S."

We keep doing this for all the pieces of "P."

1

We take "N" and add some extra stuff to it using the "pad" rule. This new string is called "P."

2

We find out how many pieces we can divide "P" into, and each piece is "r" bits long.

3

We take each piece of "P," mix it up with "S" using the "f" machine, and put the result back into "S."

4

After combining everything, we take a part of "S," chop it down to "r" bits (if it's longer), and add it to "Z."

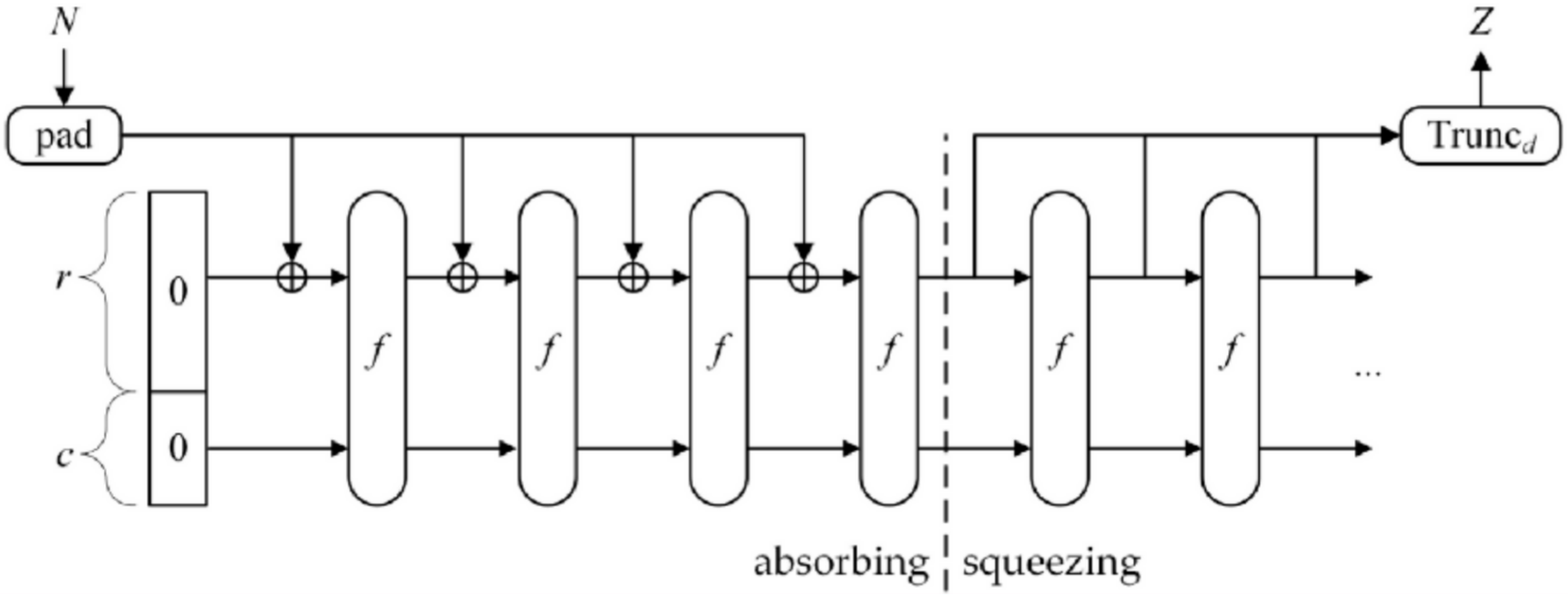
5

If "Z" is long enough (the same length as "d"), we're done and return it.

If not, we go back to step 4 and keep adding more bit

WHY "SPONGE"?

An arbitrary number of input bits are "absorbed" into the state of the function, after which an arbitrary number of output bits are "squeezed" out.



PROPERTIES OF SHA-3 [BAD]

DETERMINISTIC

- For a given input, the output (hash) will always be the same.
- This can be a concern since an attacker might gain information, even without knowing the actual input.

FAST TO COMPUTE

- For any given data, the hash can be quickly computed.
- Might enable attackers to quickly test a large number of potential inputs, compromising security.

PROPERTIES OF SHA-3 [GOOD]

AVALANCHE EFFECT

- A tiny change in the input should produce such a drastic change in output that the new hash appears uncorrelated with the old hash.
- Adds a layer of security by making it computationally infeasible to predict or deduce information about the input

COLLISION-RESISTANT

- It should be computationally difficult to find two different inputs that produce the same output.
- Enhances the reliability of the hash function

IRREVERSIBLE

- It should be computationally difficult to regenerate the original input value given the hash output
- Ensures integrity

CRYPTOGRAPHIC ATTACKS

SHA-3 IS DESIGNED TO RESIST SEVERAL TYPES OF CRYPTOGRAPHIC ATTACKS, PROVIDING A HIGH LEVEL OF SECURITY, WHILE IT FAILS TO RESIST SOME OTHER ONES

Attacks that SHA-3 can resist	Attacks that SHA-3 can’t resist
Collision Attacks	Implementation-Specific Vulnerabilities
Pre-image Attacks	Side-Channel Attacks
Cryptanalysis	Attacks on the Broader Cryptographic System
Length Extension Attacks	
Quantum Attacks (Post-Quantum Resilience)	

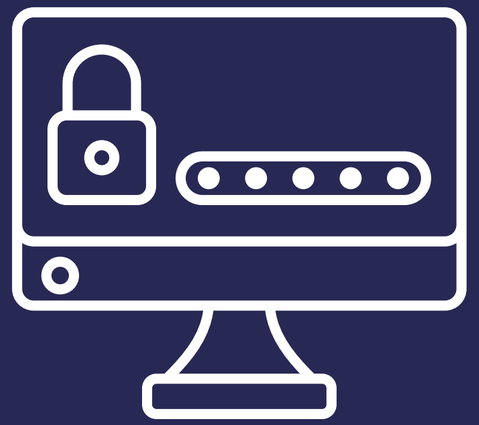
DEMOS

DEMO 1:

PASSWORD HASHING

- PASSWORDS ARE TYPICALLY HASHED USING SHA-3 BEFORE STORAGE IN A DATABASE. THIS PROCESS INVOLVES TAKING THE USER'S PASSWORD, ADDING A RANDOM "SALT" VALUE TO IT, AND THEN APPLYING SHA-3 TO GENERATE A FIXED-LENGTH HASH.
- PASSWORD HASHING IS WIDELY USED IN REAL-WORLD APPLICATIONS, PARTICULARLY IN WEBSITE AUTHENTICATION.

IN THIS DEMO WE WILL SIGN UP IN A UI, SEE THE HASHED PASSWORD IN THE DATABASE THEN LOG IN, TO SHOW HOW HASHED PASSWORDS ARE CHECKED WHILE LOGGING

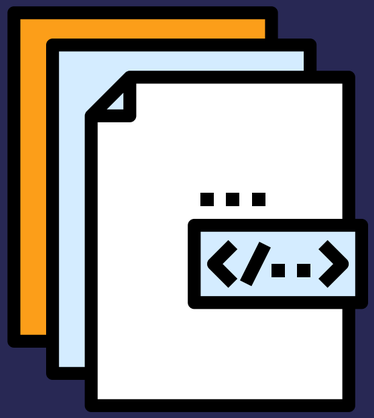


DEMO 2:

FILE DUPLICATION

- FILE DEDUPLICATION IS A DATA OPTIMIZATION TECHNIQUE THAT PLAYS A CRUCIAL ROLE IN EFFICIENTLY MANAGING AND CONSERVING STORAGE RESOURCES.
- FILE DEDUPLICATION TYPICALLY WORKS BY USING HASH FUNCTIONS,IN OUR CASE WE USED SHA-3, TO GENERATE UNIQUE FINGERPRINTS (HASHES) FOR EACH FILE. THESE FINGERPRINTS ARE COMPARED TO IDENTIFY DUPLICATES WHICH ARE REPLACED WITH REFERENCES TO THE ORIGINAL DATA

IN THIS DEMO, WE HAVE A FOLDER WITH MULTIPLE FILES SOME ARE DUPLICATES THEN WE RUN OUR CODE THAT HASHES THESE FILES AND REMOVES THE DUPLICATE COPIES FROM THE FOLDER, LEAVING US WITH ONE COPY FOR EACH FILE



DEMO 3:

MESSAGE DIGESTS

Message Digests is a string created by using SHA3 algorithm (one way function). It helps to ensure integrity of message between two parties.

In this demo, we will generate a message digest (string of digests) using MassegeDigest class in Java.

DEMO 4:

FILE CHECKSUM

File Checksum is a hash created by using SHA3 algorithm. It is used to verify if a file or data has been modified (ensure integrity). For example, if a file has been modified the file checksum will be changed, and then we will know if the file has been modified or it is not the same as original file.

In this demo, we will create a hash values for a file using MessageDigest class in Java. First we will read the data of the file, then we will generate a hash for it using SHA3 algorithm.

DEMO 5:

DIGITAL SIGNATURES

- DIGITAL SIGNATURES ENSURE AUTHENTICITY AND NON-REPUDIATION IN TERMS OF SECURING VARIOUS TYPES OF DATA.
- SHA-3 IN DIGITAL SIGNATURES, SIGNIFICANTLY REDUCES THE CHANCES OF FORGERY OR IMPERSONATION.

FOR THIS DEMONSTRATION, WE WILL UTILIZE THE SHA-3 AS WELL AS THE DIGITAL SIGNATURE ALGORITHM, TO SHOWCASE THE FUNCTIONALITY OF DIGITAL SIGNATURES.

WE WILL INITIALLY CREATE, ADD CONTENT, SIGN A DOCUMENT AND THEN PROCEED TO MAKE MODIFICATIONS TO IT, TO PROVE THE DETECTION OF MODIFICATIONS BY ANAUTHORIZED USERS.

THE VERIFICATION IS THE RESULT OF THE COMPARISON OF THE DOCUMENT'S HASH, BEFORE AND AFTER MODIFICATION.

CONCLUSION

IN CONCLUSION:

- SHA-3 stands as a groundbreaking achievement in the realm of cryptographic hash functions.
- SHA-3 operates on innovative sponge construction, which offers a robust foundation for secure data processing.
- When compared to SHA-2, SHA-3 distinguishes itself through its unique design principles and enhanced security features.
- SHA-3's practical utility is exemplified through its role in real-world scenarios.
- SHA-3 represents a significant milestone in safeguarding digital assets and maintaining trust in data transactions.

THANK YOU FOR LISTENING!