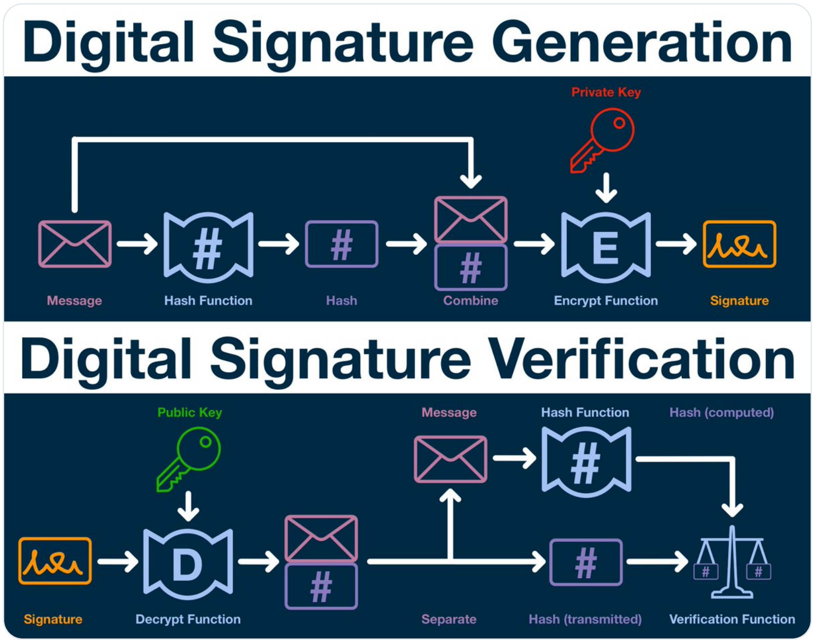
Haym @SalomonCrypto
Nov 17 • 23 tweets • SalomonCrypto/status/1593040051402784768

(1/22) Cryptography Fundamentals: Digital Signatures

The internet is the Wild West; how can our world become increasingly reliant on a system with no authorities... where the only laws that matter are the laws of physics and code?

A guide to (mathematically) trustless trust.



(2/22) Cryptography is the study of techniques to protect information so that only people can access it are those for whom the information is intended.

Encryption coverts information into a secret code that hides its true meaning; decryption moves the other way.

(3/22) There are two types of encryption: symmetric and asymmetric.

Symmetric encryption uses the same set of keys to encrypt and decrypt a message.

Think: "Alice and Bob share a secret number (key) that they use to both encrypt and decrypt messages."

(4/22) Asymmetric encryption uses different keys for encryption and decryption.

Think: "Alice has two keys, encryption and decryption. She shares the encryption key with anyone, who can then use it to create secret messages which can only be read with the decryption key."

(5/22) Asymmetric encryption is powerful because it can be used without a shared secret. While creating a shared secret isn't that challenging (eg using Diffie-Hellman), it still requires a non-trivial amount of work.

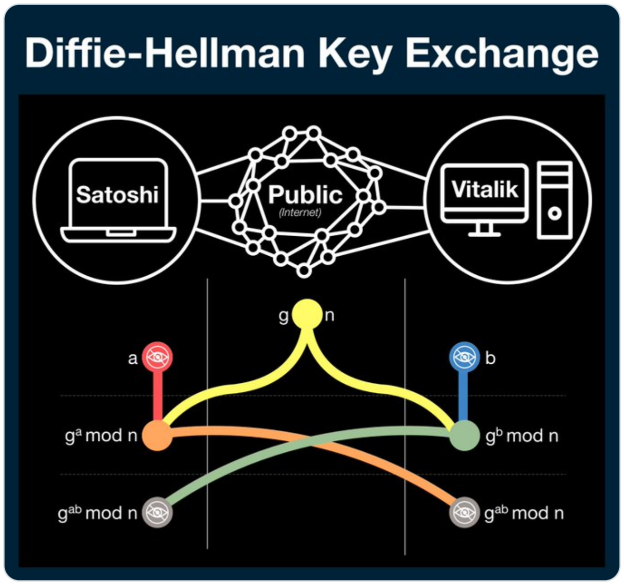
Both parties must actively interact to create the secret.

Haym
@SalomonCrypto · Follow

(1/21) Cryptography Fundamentals: Diffie-Hellman Key Exchange

How do you share private information over public networks? How can we create mathematically secure secrets? What actually is an encryption key?

A public guide for private communication.



2:43 AM · Oct 13, 2022

Read the full conversation on Twitter

461 Reply Copy link

Read 29 replies

(6/22) For some applications, this is ok. If you are using encryption to transfer sensitive data to your bank, it's reasonable for both parties to engage.

For others, this is a huge problem. The reality is that both parties will not always be available to generate a secret.

(7/22) We are particularly concerned with a specific application of encryption: digital signatures.

A digital signature is used to publicly confirm two things:

1) the message was really sent by the person claiming to send it
2) the message has not been tampered with

(8/22) At the highest level, the purpose of digital signatures is to mathematically prove that the underlying data has not been tampered, modified or otherwise changed.

Using a signature anyone can verify a message, even if the signatory is no longer active.

(9/22) There are two industry-standard ways to create digital signatures: RSA (Rivest-Shamir-Adleman) and DSA (digital signature algorithm).

We'll get into implementation details in another thread, but both algorithms work by the same general principle.

(10/22) First, our cryptographic signer needs to generate both his signing and verification keys.

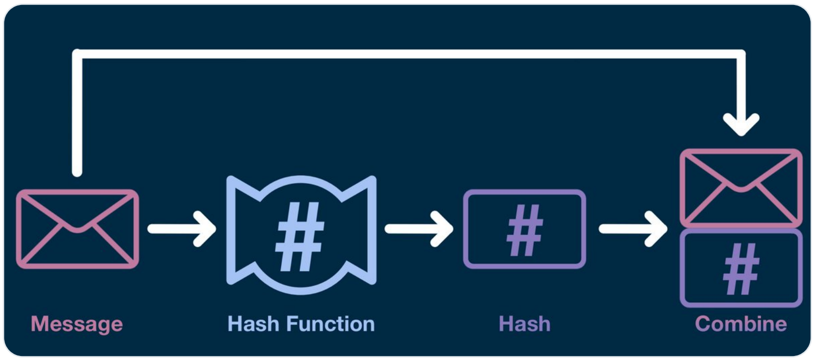
The signing key is used to create signatures and must remain completely PRIVATE; anyone with access can create valid signatures.

The verification key can be PUBLICly distributed.

(11/22) Now the signer must prepare the message for encryption/signature.

The message (the information being signed) is passed through hash function, generating a message hash.

Then, the hash is appended to the end of the original message.



(12/22) <NOTE>

Hash function: a one way function that takes in arbitrary data and outputs a unique, random-seeming string. It is very easy to calculate, but nearly impossible to reverse.

Imagine a hash like the data-version of a fingerprint.

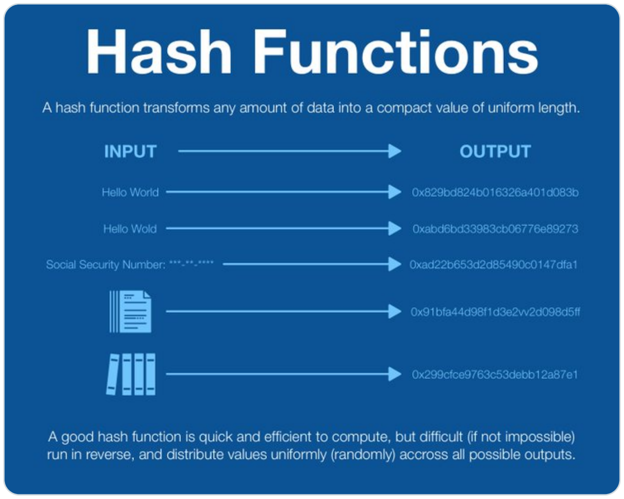
</ NOTE>

Haym
@SalomonCrypto · **Follow**

(1/7) Computer Science 101: Hash Functions

What is a hash function? What are the characteristics of a good hash function? Where do hash functions appear and why do I hear about them all the time?

If you want to understand the fundamental tool of crypto, this guide is for you!


Hash Functions
A hash function transforms any amount of data into a compact value of uniform length.
INPUT
OUTPUT
Hello World
0x829bd824b016326a401d083b
Hello Wold
0xabd6bd33983cb06776e89273
Social Security Number: ***-**-****
0xad22b653d2d85490c0147dfa1
0x91bfa44d98f1d3e2vv2d098d5ff
0x299cfce9763c53debb12a87e1
A good hash function is quick and efficient to compute, but difficult (if not impossible) run in reverse, and distribute values uniformly (randomly) accross all possible outputs.

3:54 PM · Sep 7, 2022

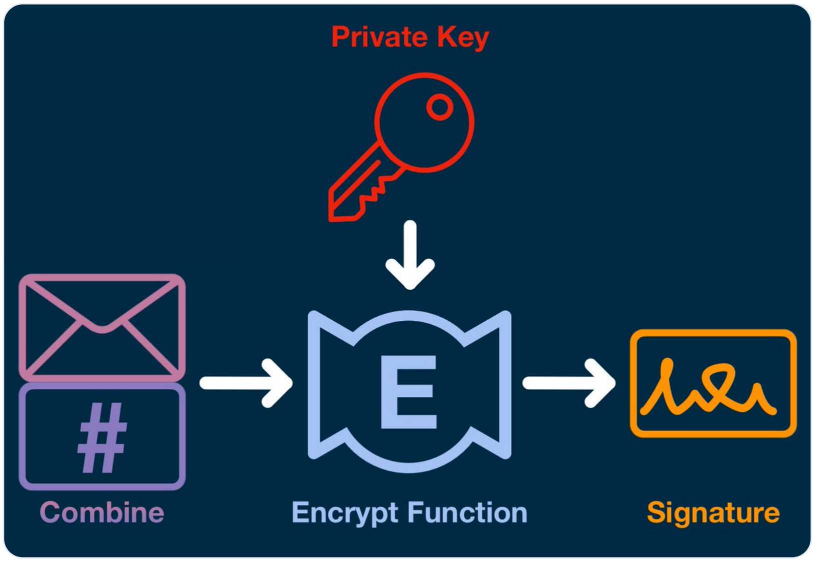
Read the full conversation on Twitter

132 Reply Copy link

Read 1 reply

(13/22) Once the signer has prepared and appended the hash, he is ready to sign.

The combined message-hash and the signer's private key are fed into the encrypt function. The encrypt function will combine both components using advanced math to create the digital signature.



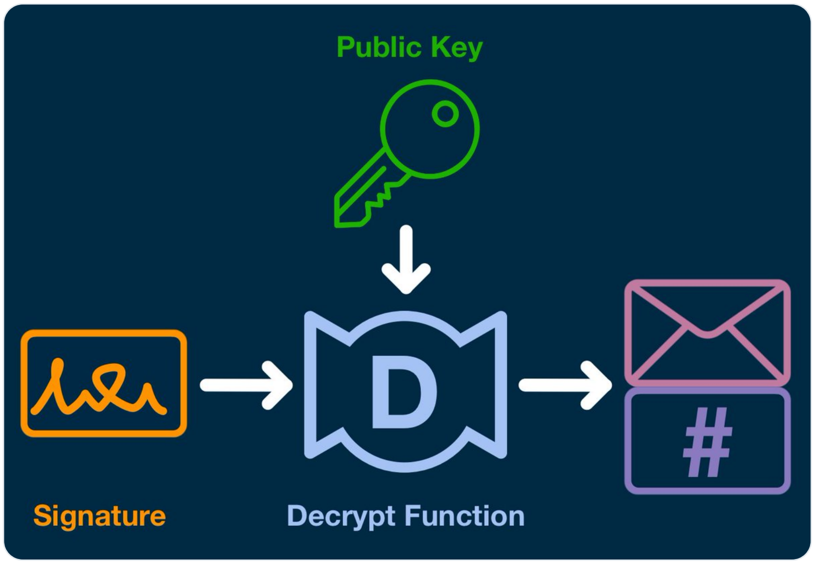
(14/22) That's the entire signature process! The encrypt function will output the signature data, which can be freely and publicly shared.

The signature will not leak any of the underlying message or private key data. It will just look like a random, meaningless chunk of data

(15/22) Now it is time to verify the signature.

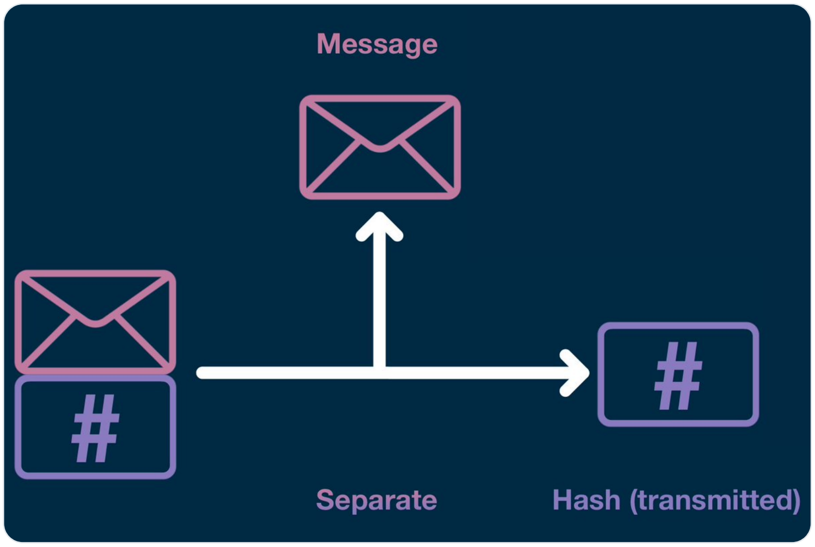
To begin, we need to undo the encryption process, which transformed our combined message-hash into a single piece of opaque data.

And so, the verifier feeds the signature and the signers public key into the decryption function.



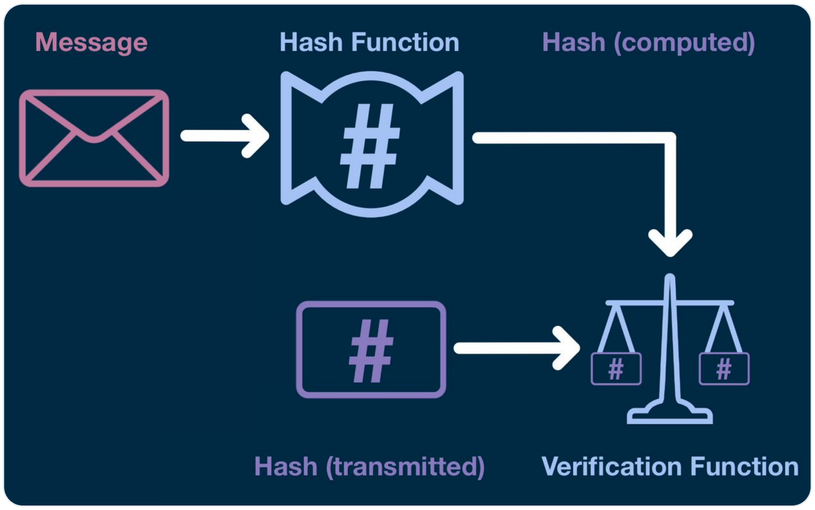
(16/22) Assuming the signer created the signature with a private key that corresponds to that public key, the signature will decrypt into the same message-hash used to create the signature.

If the wrong key is used, the signature will decrypt into gibberish.



(17/22) If the decryption was successful, the verifier can then pull apart the message-hash, take the message portion and run it through the hash function.

Finally, the verifier will feed this computed hash and the (decrypted) transmitted hash into the verification function.



(18/22) The verification function is the final step of the digital signature process. If the message message was signed by the sender who generated the public keys, the verification function will prove it to you with mathematical certainty.

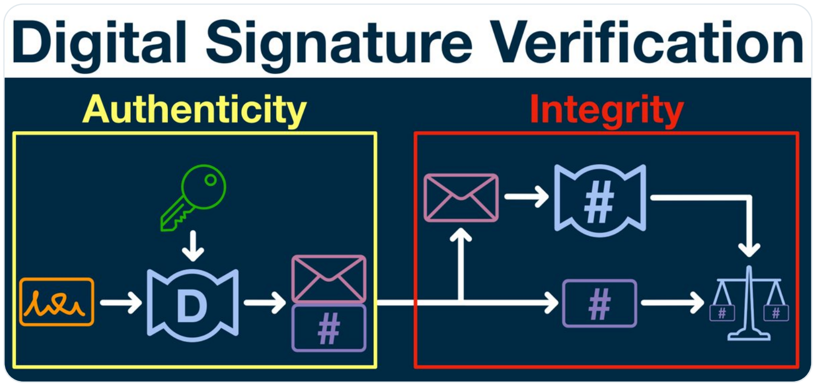
(19/22) Before we wrap, let's return to the goals of digital signatures in tweet 7:

1) the message was really sent by the person claiming to send it (authenticity)
2) the message has not been tampered with (integrity)

(20/22) Now let's take a second look at digital signature verification, this time as a two-step process.

First, we decrypt the signature - only possible if we have the correct public key.

Second, we confirm the message is correct - only possible if the message matches the hash.



(21/22) Finally, a few words on implementation. As previously mentioned, there are two main digital signature algorithms: RSA and DSA.

Both are considered mathematically safe, but DSA is faster at decrypting and signing, while RSA is faster at encrypting and verifying.

(22/22) As you will see in future threads on RSA and DSA, the differences appear (mostly) in the encrypt, decrypt and verification function.

But, in general, all digital signature schemes follow the basic principles outlined in this thread.

More of a long-form reader? Try this:

Haym
@SalomonCrypto

Cryptography Fundamentals: Digital Signatures

Cryptography Fundamentals: Digital Signatures | Haym
(1/22) Cryptography Fundamentals: Digital Signatures The internet is the Wild West; how can our world become increasingly reliant on a system with no authorities... where the only laws that matter a…
https://typefully.com/SalomonCrypto/aQDRhyx

Like what you read? Help me spread the word by retweeting the thread (linked below).

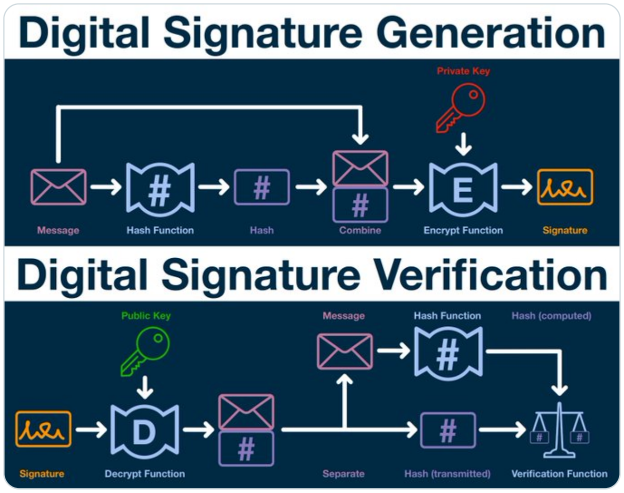
Follow me for more explainers and as much alpha as I can possibly serve.

Haym
@SalomonCrypto · **Follow**

(1/22) Cryptography Fundamentals: Digital Signatures

The internet is the Wild West; how can our world become increasingly reliant on a system with no authorities... where the only laws that matter are the laws of physics and code?

A guide to (mathematically) trustless trust.



12:35 AM · Nov 17, 2022

Read the full conversation on Twitter

229 Reply Copy link

Read 9 replies

• • •