PKI

PKI - public key infrastructure

Signer
- Kasutades x hash algoritmi, nt SHA luuakse andmete/data põhjal hash
- Kasutades privaatset võtit ja x krüptograafia algoritmi, nt RSA, loodud hash krüpteeritakse
- Kirja pannakse ka täpne aeg, millal hash loodi/krüpteeriti ehk ajatempel
- Kaasa pannakse sertifikaat, mis tõestab, et antud privaatse võtme omanik on see, kes ta väidab, et ta on. Samuti sisaldab sertifikaat signeerija avalikku võtit
- Sertifikaat saadakse nt CA avalikust LDAP-st
- Originaal andmetega/dataga pannakse kaasa 1) krüpteeritud hash 2) täpne aeg 3) sertifikaat

Verifier
- Kasutades avalikku/pub võtit sertifikaadi seest ja x krüptograafia algoritmi, edastatud hash dekrüpteeritakse
- Kasutades x hash algoritmi, luuakse andmete/data põhjal hash
- Võrreldakse andmete/data põhjal loodud hashi ja dekrüpteeritud hashi. Kui need on samad, on allkiri õige ehk esialgset datat/andmeid ei ole muudetud

Eestis signeerides id-kaardi, m-id või smart-id’ga võetakse isiku sertifikaat avalikust LDAP-st kuna sertifikaat ei sisalda isiklikku ega tundliku infot.

PKI involves using a digital certificate for identity verification.

Root cert
Intermediate cert
End user/entity cert

Certificate
The certificate is used to confirm that the public key belongs to the specific organization or identity.
- issued by a CA
- contains the public key for a digital signature and specifies the identity associated with the key, such as the name of an organization
- CA acts as the guarantor
- Digital certificates must be issued by a trusted authority and are only valid for a specified time. They are required in order to create a digital signature
-

SHA - hash algo
one way function, turning an input of any size into a fixed-length output

RSA - encrypt algo
encrypting an input into an output that can then be decrypted

It uses a different key for encryption (the public one) than for decryption (the private one). This can therefore be used to receive encrypted messages from others - you can publish your public key, but only you with the private key can then decrypt the messages that have been encrypted with it.

If you reverse the keys for RSA, it can be used to generate a digital signature - by encrypting something with your private key, anyone can decrypt it with the public key and, if they are sure the public key belongs to you, then they have confidence that you were the one who encrypted the original. This is normally done in conjunction with a hash function - you hash your input, then encrypt that with your private key, giving a digital signature of a fixed length for your input message.

One more note is that hash algorithms, like SHA-1, can compute digests given data of any length as input. Asymmetric algorithms, like RSA, are limited in the length of data they can transform. For that reason, the original message is rarely signed with RSA, and instead the SHA-1 digest of the original message is signed. The recipient of the message and its signature computes the SHA-1 digest of the message, then decrypts the signature with the sender's public key and verifies that the digests exactly match

The keys work inversely to each other. Encrypted something with your public key? Decrypt it with your private key. Conversely, if you encrypted something with your private key, you decrypt it with your public. Such is the nature of asymmetric cryptography.

Encryption with the private key is used to prove authenticity. If person 1 encrypts a message with their own private key then person 2 can decrypt it with person 1's public key, which proves that person 1 originated the message since it could only have been encrypted with their private key.

https://www.docusign.com/how-it-works/electronic-signature/digital-signature/digital-signature-faq
https://stackoverflow.com/questions/733692/sha1-vs-rsa-whats-the-difference-between-them

---

Revision #2
Created 2024-12-27 20:17:11 UTC by Admin
Updated 2026-02-05 11:36:10 UTC by qrl