

PKI

PKI

Signer

- kasutades x hash algot, nt SHA luuakse data põhjal hash
- kasutades priv võtit ja x krüto algot, nt RSA, loodud hash krüpteeritakse
- kirja pannakse ka täpne aeg, millal hash loodi/krüpteeriti ehk ajatempel
- kaasa pannakse sertifikaat, mis tõestab, et antud priv võtme omanik on see kes ta väidab, et on. samuti sisaldab sert signeerija avalikku võtit
- sert saadakse nt CA avalikust LDAP-st
- originaal dataga pannakse kaasa 1) krüptitud hash 2) täpne aeg 3) sert

Verifier

- kasutades pub võtit serdi seest ja x krüpto algot, edastatud hash dekrüpteeritakse
- kasutades x hash algot luuakse data põhjal hash
- võrreldakse data põhjal loodud hashi ja dekrüpteeritud hashi. kui need kattuvad, on allkiri õige ehk esialgset datat ei ole muudetud

Eestis signeerides id-kaardi, m-id või smart-id'ga võetakse isiku sert avalikust LDAP-st kuna sert 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 #1

Created 27 December 2024 20:17:11 by Admin

Updated 27 December 2024 20:17:16 by Admin