Nordea

PKI and related challenges

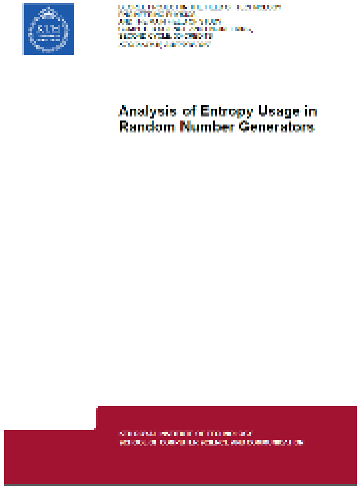
Hannes Salin, IT Senior Developer and Scrum Master
8.1.2019

Agenda

- My work and challenges with PKI
- PKI for developers
- Microservices and PKI
- Alternatives to PKI

Hannes Salin

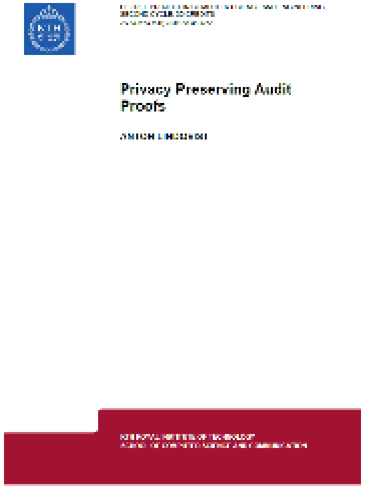
- Scrum master and developer at Nordea
- Authentication and signing applications
- Master thesis supervisor and project manager
- DFS Dalarna network leader IT-security
- Researching
- Writing whitepapers
- Lecturing on cryptography and related fields


Analysis of Entropy Usage in Random Number Generators

A secure multi-party scheme with certificateless cryptography for secret key extraction

DENNIS FORTH

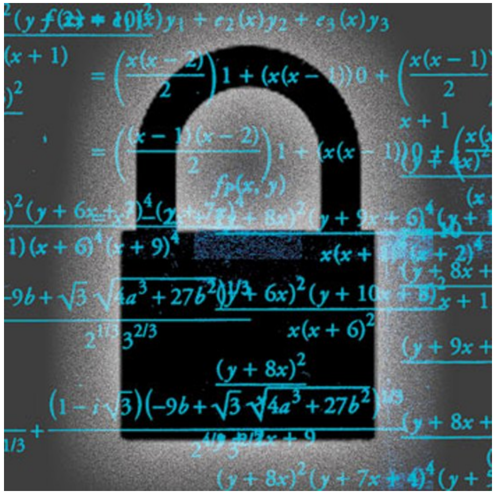
Providing Secure Randomness in Applied Cryptography


Privacy Preserving Audit Proofs

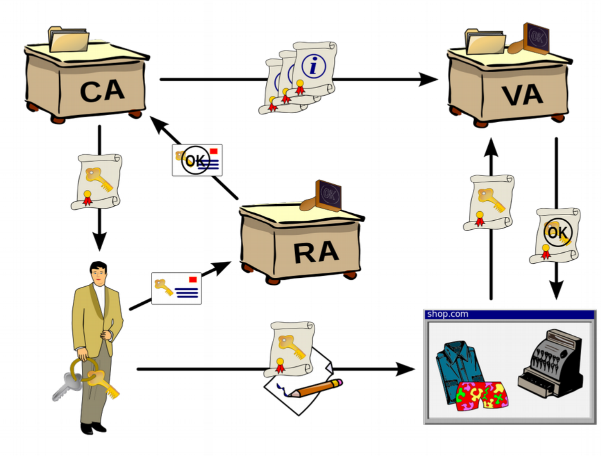
ANTON [illegible]

My work (and challenges) with PKI

- Relationship between Public Key Cryptography (PKC) and Public Key Infrastructure (PKI)
- PKC is collection of asymmetric crypto schemes (e.g. RSA, DH) with two main operations:
  - encrypt – decrypt (key exchange / encryption)
  - sign – verify (digital signature)
- PKI provide to users some verifiable guarantee as to the ownership of public keys
- PKC need PKI to solve key distribution issue
- PKI uses PKC to execute operations needed, e.g. digital signatures

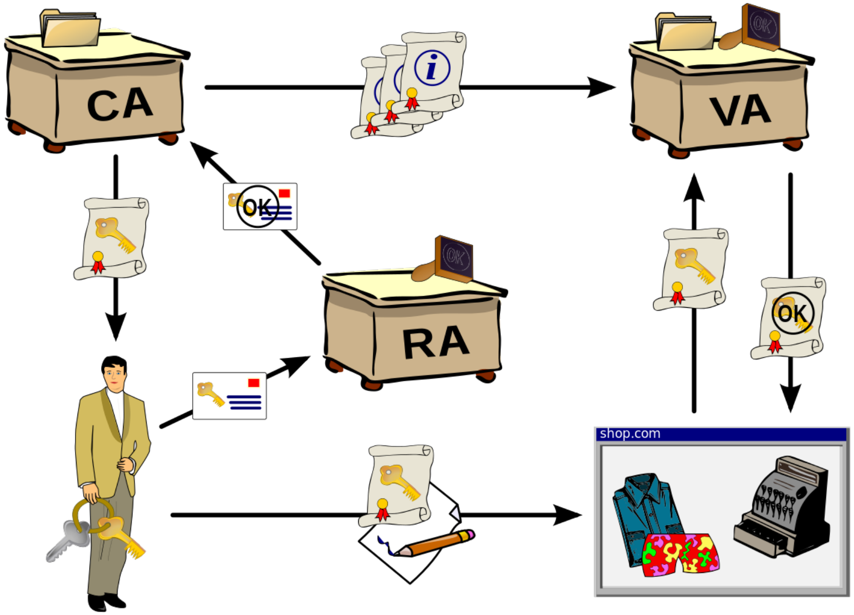


≡?



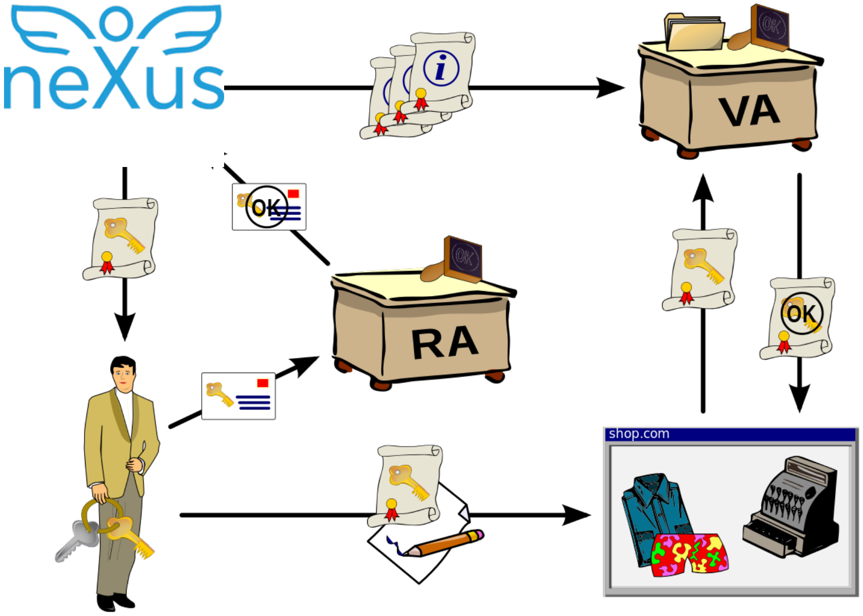
My work (and challenges) with PKI

- First challenge: understanding PKI and all components



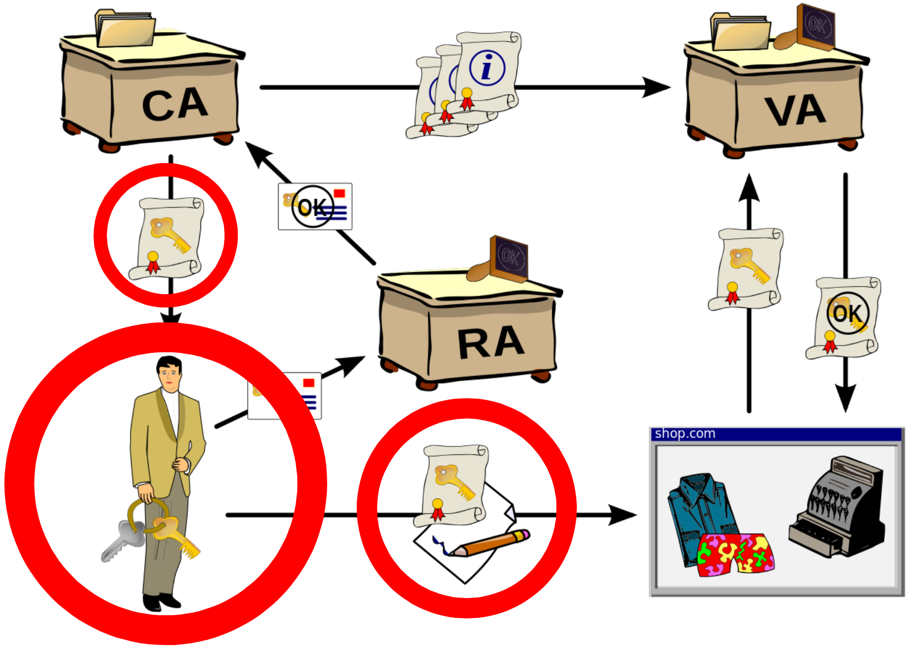
My work (and challenges) with PKI

- First challenge: understanding PKI and all components
- Second challenge: securing the life-cycle of certificates – worked on Nexus Certificate Manager (also PRIME and Hybrid Access Gateway)



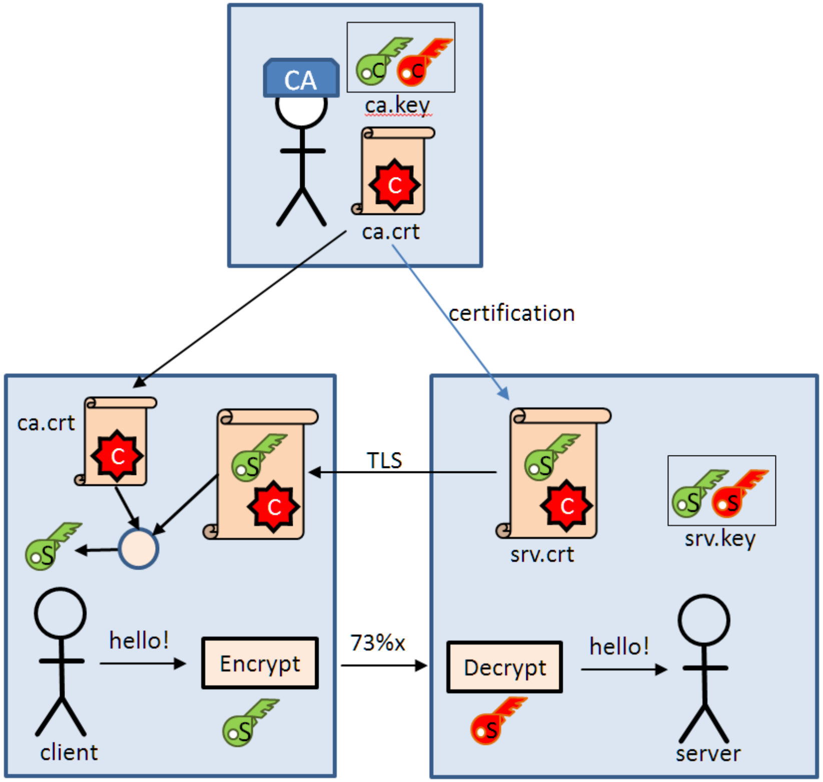
My work (and challenges) with PKI

- First challenge: understanding PKI and all components
- Second challenge: securing the life-cycle of certificates – worked on Nexus Certificate Manager, PRIME and Hybrid Access Gateway
- Third challenge: setup proper PKI in secure financial systems



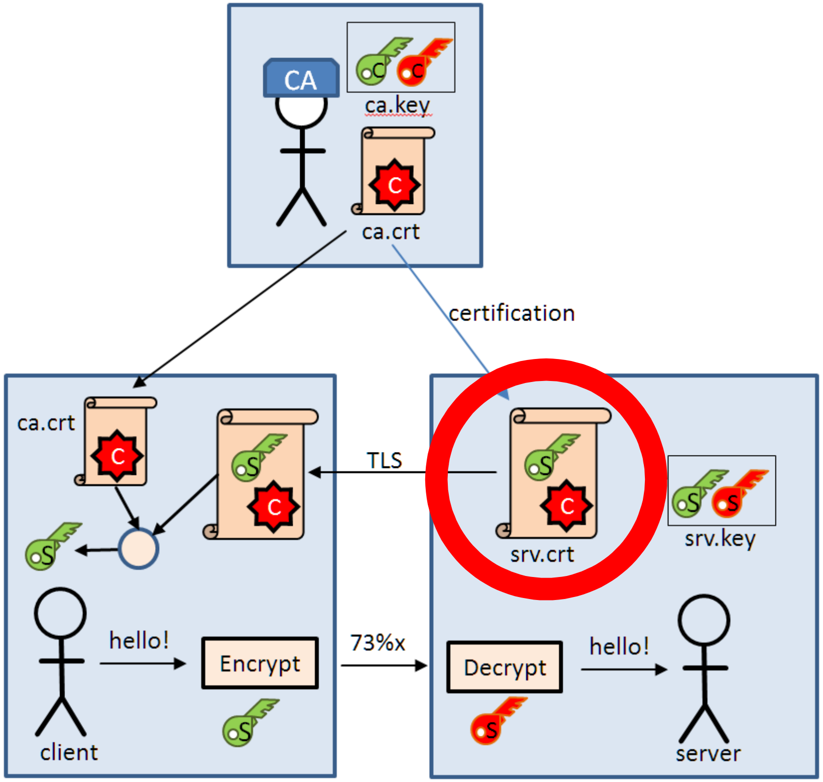
My work (and challenges) with PKI

- First challenge: understanding PKI and all components
- Second challenge: securing the life-cycle of certificates – worked on Nexus Certificate Manager, PRIME and Hybrid Access Gateway
- Third challenge: setup proper PKI in secure financial systems
- Fourth challenge: the world of TLS



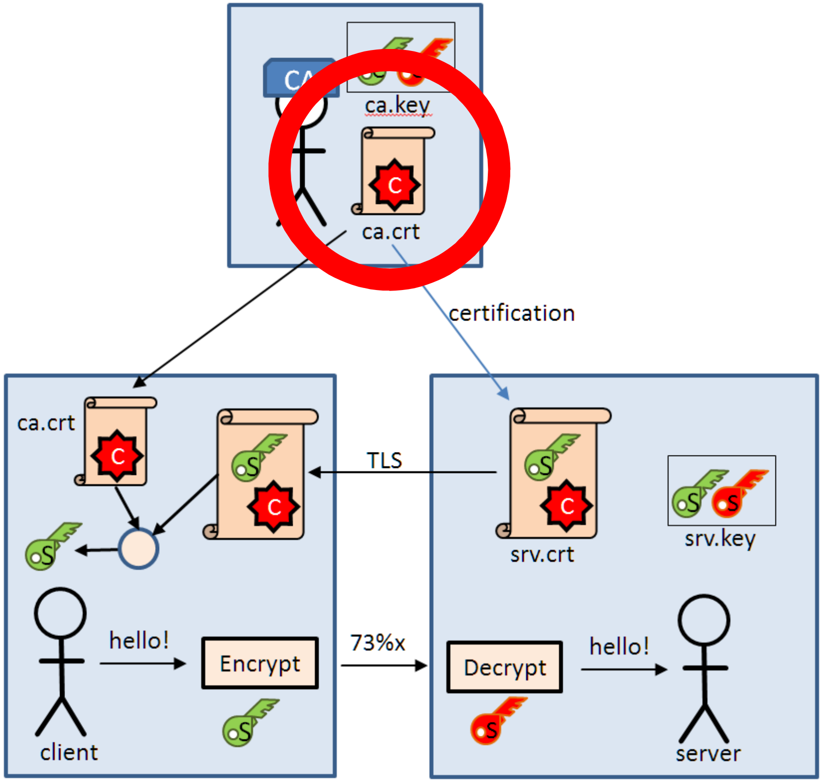
My work (and challenges) with PKI

- First challenge: understanding PKI and all components
- Second challenge: securing the life-cycle of certificates – worked on Nexus Certificate Manager, PRIME and Hybrid Access Gateway
- Third challenge: setup proper PKI in secure financial systems
- Fourth challenge: the world of TLS
  - Expired certificates



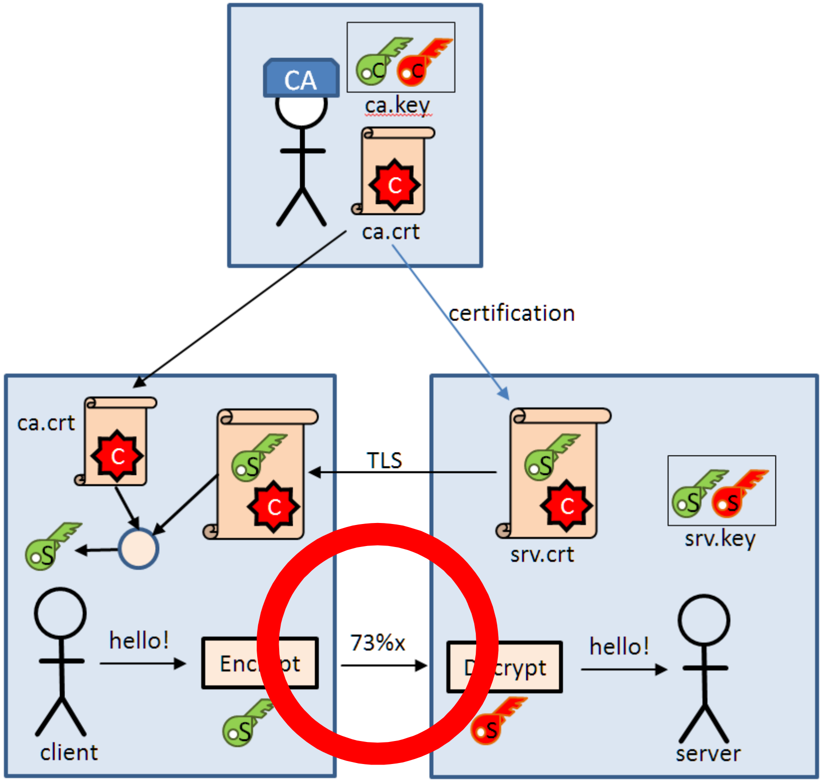
My work (and challenges) with PKI

- First challenge: understanding PKI and all components
- Second challenge: securing the life-cycle of certificates – worked on Nexus Certificate Manager, PRIME and Hybrid Access Gateway
- Third challenge: setup proper PKI in secure financial systems
- Fourth challenge: the world of TLS
  - Expired certificates
  - Compromised CA



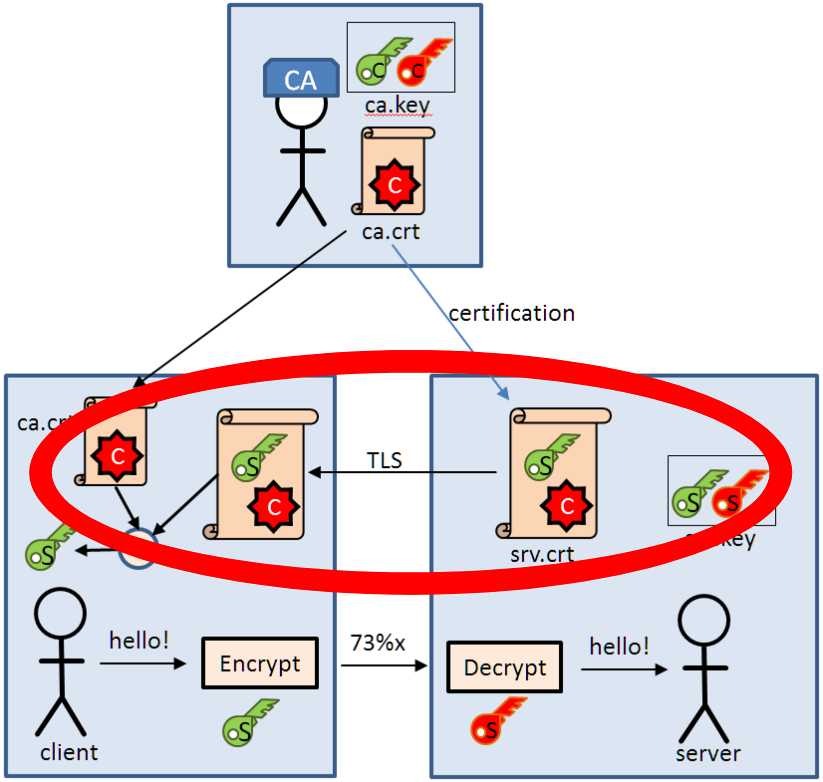
My work (and challenges) with PKI

- First challenge: understanding PKI and all components
- Second challenge: securing the life-cycle of certificates – worked on Nexus Certificate Manager, PRIME and Hybrid Access Gateway
- Third challenge: setup proper PKI in secure financial systems
- Fourth challenge: the world of TLS
  - Expired certificates
  - Compromised CA
  - Faulty TLS configurations



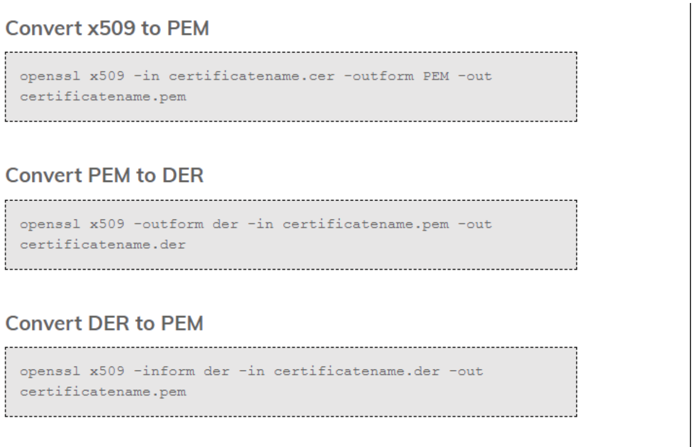
My work (and challenges) with PKI

- First challenge: understanding PKI and all components
- Second challenge: securing the life-cycle of certificates – worked on Nexus Certificate Manager, PRIME and Hybrid Access Gateway
- Third challenge: setup proper PKI in secure financial systems
- Fourth challenge: the world of TLS
  - Expired certificates
  - Compromised CA
  - Faulty TLS configurations
- Fifth challenge: Segregation of Duties



PKI for developers

- Needed level of knowledge?
- Certificates and TLS is the practical part of PKI a developer get contact with
  - Truststore / Keystore
  - Key- and certificate formats (and conversion between them)
  - Cipher suites
  - Certificate pinning
  - **Java’s keytool and OpenSSL are good tools to know**
- Revocations and certificate signing requests is more on operational side



Convert x509 to PEM

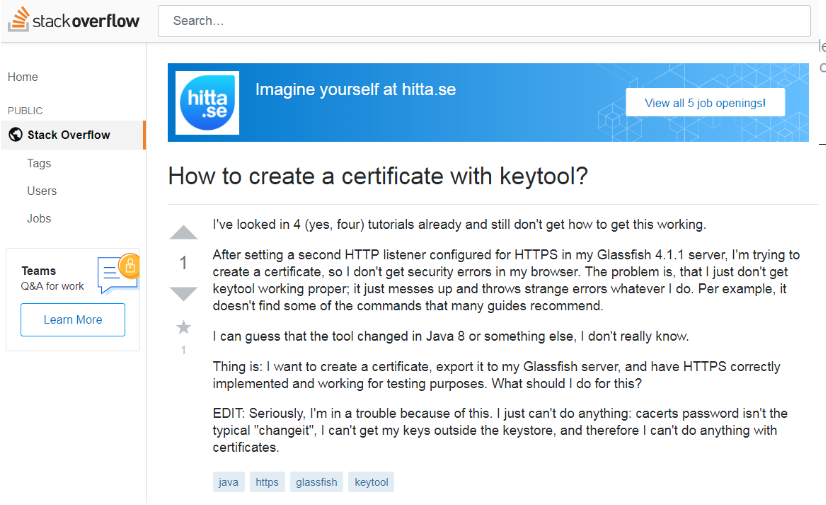
```
openssl x509 -in certificatename.cer -outform PEM -out
certificatename.pem
```

Convert PEM to DER

```
openssl x509 -outform der -in certificatename.pem -out
certificatename.der
```

Convert DER to PEM

```
openssl x509 -inform der -in certificatename.der -out
certificatename.pem
```



How to create a certificate with keytool?

1

I've looked in 4 (yes, four) tutorials already and still don't get how to get this working.

After setting a second HTTP listener configured for HTTPS in my Glassfish 4.1.1 server, I'm trying to create a certificate, so I don't get security errors in my browser. The problem is, that I just don't get keytool working proper; it just messes up and throws strange errors whatever I do. Per example, it doesn't find some of the commands that many guides recommend.

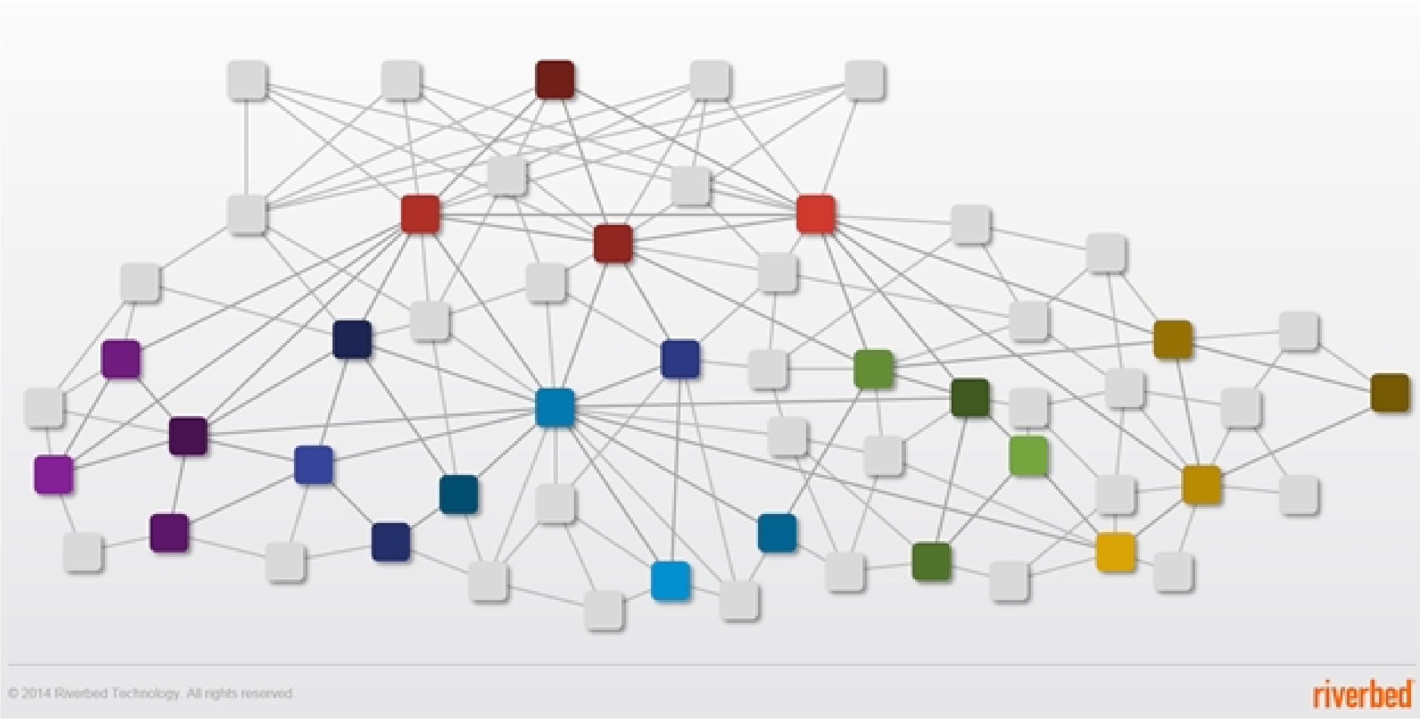
I can guess that the tool changed in Java 8 or something else, I don't really know.

Thing is: I want to create a certificate, export it to my Glassfish server, and have HTTPS correctly implemented and working for testing purposes. What should I do for this?

EDIT: Seriously, I'm in a trouble because of this. I just can't do anything: cacerts password isn't the typical "changeit", I can't get my keys outside the keystore, and therefore I can't do anything with certificates.

Microservices and PKI

- Assume an eco-system with 30-40 microservices and a policy saying all nodes need TLS – is it manageable?
- Is it necessary with PKI within a “secure zone” or internal network?

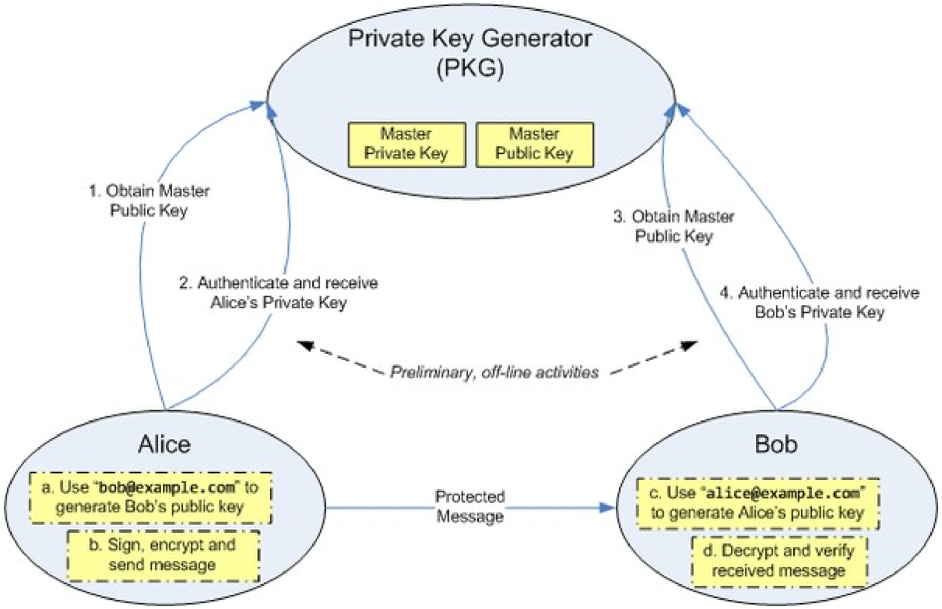


Alternatives to traditional PKI

- “…significant overhead is associated with managing digital certificates” yepp!

  “…the new notion called “identity-based public key cryptography” (ID-PKC) in which bitstring of user identity (could be name, email addresses, etc) is directly being the public key” [1]

- Private Key Generator (PKG) is the weak point
- The most efficient identity-based encryption schemes are currently based on bilinear pairings on elliptic curves, such as the Weil or Tate pairings.


Private Key Generator (PKG)
Master Private Key
Master Public Key
1. Obtain Master Public Key
2. Authenticate and receive Alice's Private Key
3. Obtain Master Public Key
4. Authenticate and receive Bob's Private Key
Preliminary, off-line activities
Alice
a. Use "bob@example.com" to generate Bob's public key
b. Sign, encrypt and send message
Protected Message
Bob
c. Use "alice@example.com" to generate Alice's public key
d. Decrypt and verify received message

[1] Survey on certificateless public key cryptography, 2011, Al Housani et al.

Nordea

Thank you!

Hannes Salin, IT Senior Developer and Scrum Master
hannes.salin@nordea.com