

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



SWEDISH INSTITUTE OF INFORMATION TECHNOLOGY
ANALYSIS OF ENTROPY USAGE IN
RANDOM NUMBER GENERATORS
HANNES SALIN
M.Sc. THESIS

Analysis of Entropy Usage in Random Number Generators



SWEDISH INSTITUTE OF INFORMATION TECHNOLOGY
SECURE CYBER IN LINE WITH
PROVISIONAL REGULATIONS

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

DANIEL FORN



Providing Secure Randomness in Applied Cryptography

Hannes Salin
M.Sc. Thesis
SICS

Abstract: Randomness is a critical element in many cryptographic applications. This thesis analyzes the entropy usage in random number generators (RNGs) used in applied cryptography. The study focuses on the entropy requirements for different types of RNGs and the impact of entropy usage on the security of cryptographic systems. The thesis also discusses the challenges of providing secure randomness in a distributed environment and presents a solution based on a secure multi-party scheme.

1. Introduction

The security of many cryptographic applications depends on the quality of the randomness used. In this thesis, we analyze the entropy usage in different types of RNGs and the impact of entropy usage on the security of cryptographic systems. The thesis also discusses the challenges of providing secure randomness in a distributed environment and presents a solution based on a secure multi-party scheme.

2. Background

Randomness is a critical element in many cryptographic applications. This thesis analyzes the entropy usage in random number generators (RNGs) used in applied cryptography. The study focuses on the entropy requirements for different types of RNGs and the impact of entropy usage on the security of cryptographic systems. The thesis also discusses the challenges of providing secure randomness in a distributed environment and presents a solution based on a secure multi-party scheme.

3. Methodology

The methodology of this thesis involves a combination of theoretical analysis and practical experimentation. We analyze the entropy requirements for different types of RNGs and the impact of entropy usage on the security of cryptographic systems. We also implement and evaluate a secure multi-party scheme for providing secure randomness in a distributed environment.

4. Results and Discussion

The results of this thesis show that the entropy requirements for different types of RNGs are significantly higher than what is currently assumed. This has implications for the security of cryptographic systems that use these RNGs. The thesis also shows that a secure multi-party scheme can be used to provide secure randomness in a distributed environment.

5. Conclusion

This thesis provides a detailed analysis of the entropy usage in random number generators used in applied cryptography. The results show that the entropy requirements for these RNGs are significantly higher than what is currently assumed. This has implications for the security of cryptographic systems that use these RNGs. The thesis also shows that a secure multi-party scheme can be used to provide secure randomness in a distributed environment.



SWEDISH INSTITUTE OF INFORMATION TECHNOLOGY
SECURE CYBER IN LINE WITH
PROVISIONAL REGULATIONS

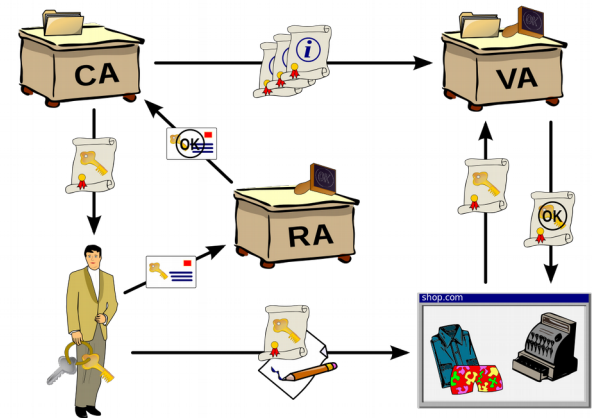
Privacy Preserving Audit Proofs

JONIK LINDHEIM



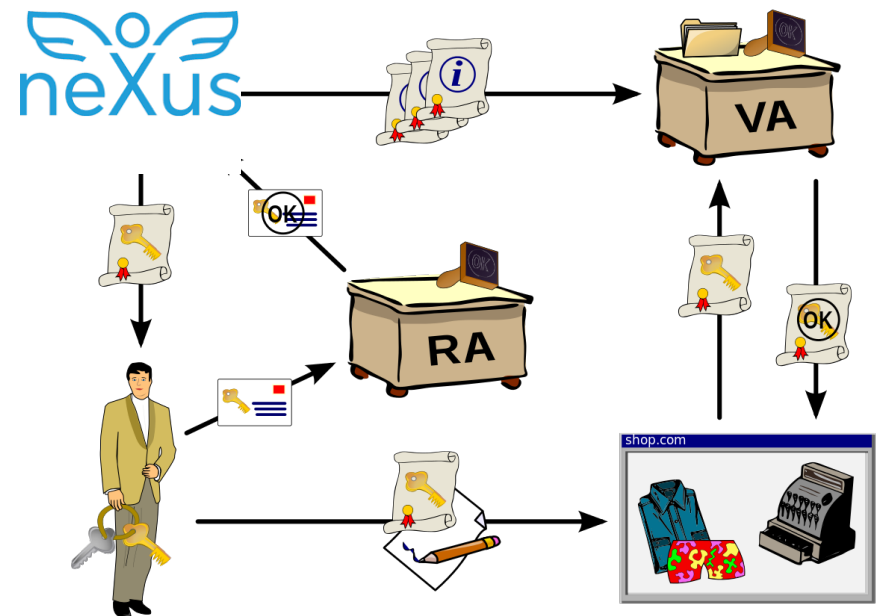
My work (and challenges) with PKI

- 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



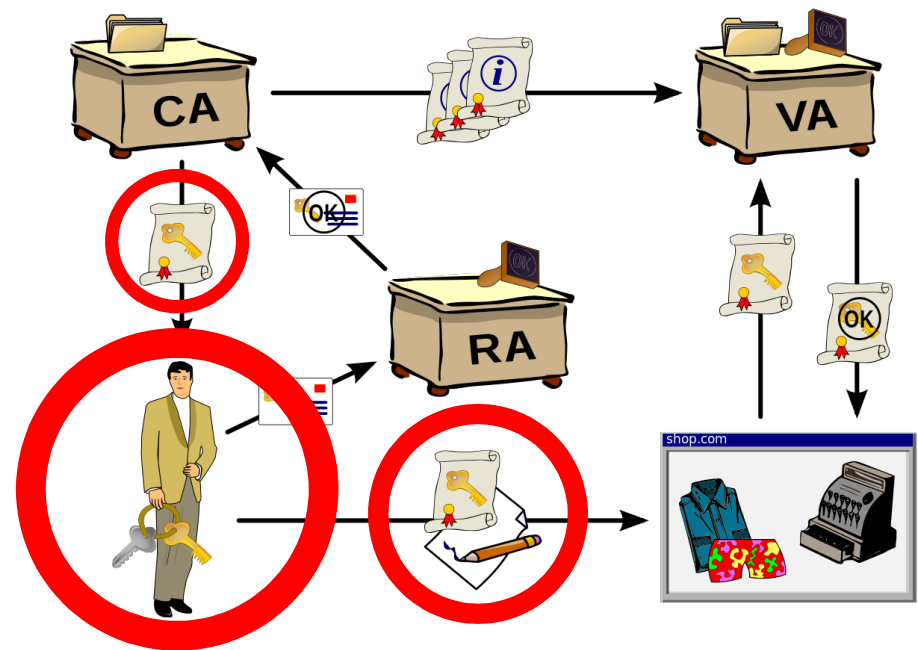
My work (and challenges) with PKI

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



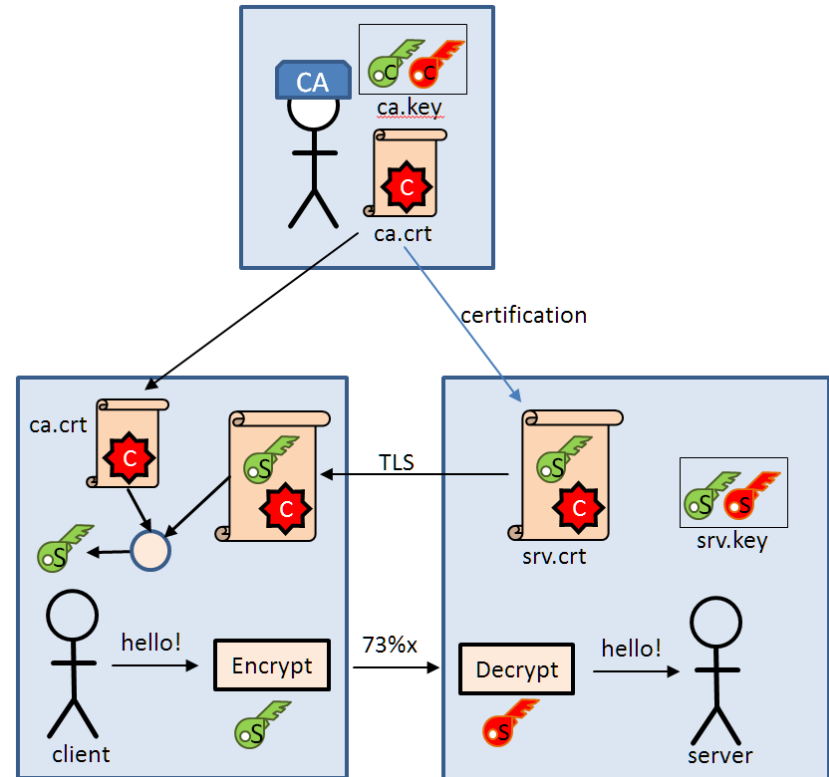
My work (and challenges) with PKI

- 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



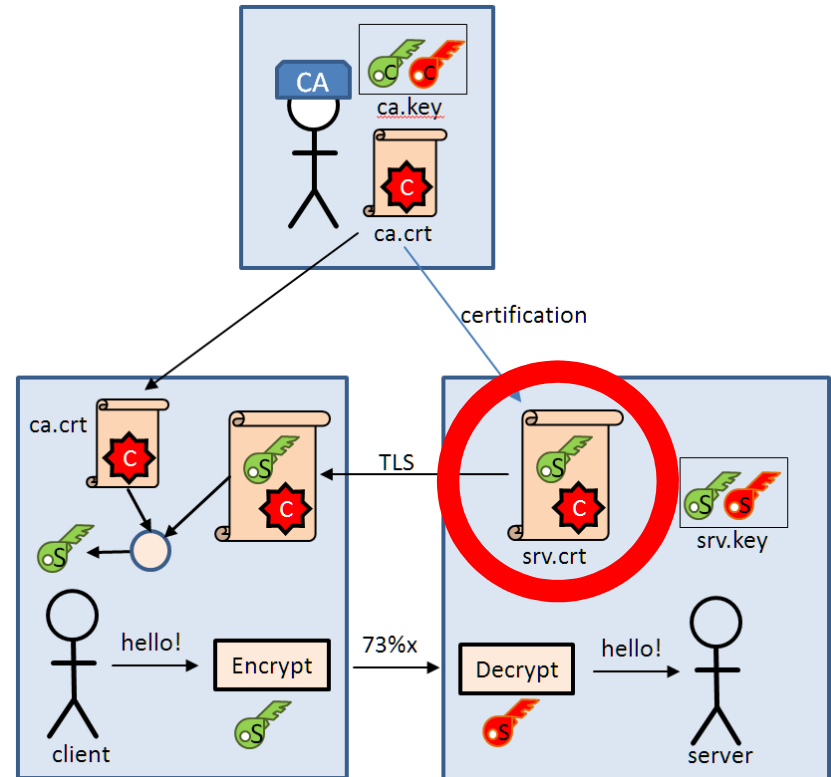
My work (and challenges) with PKI

- 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



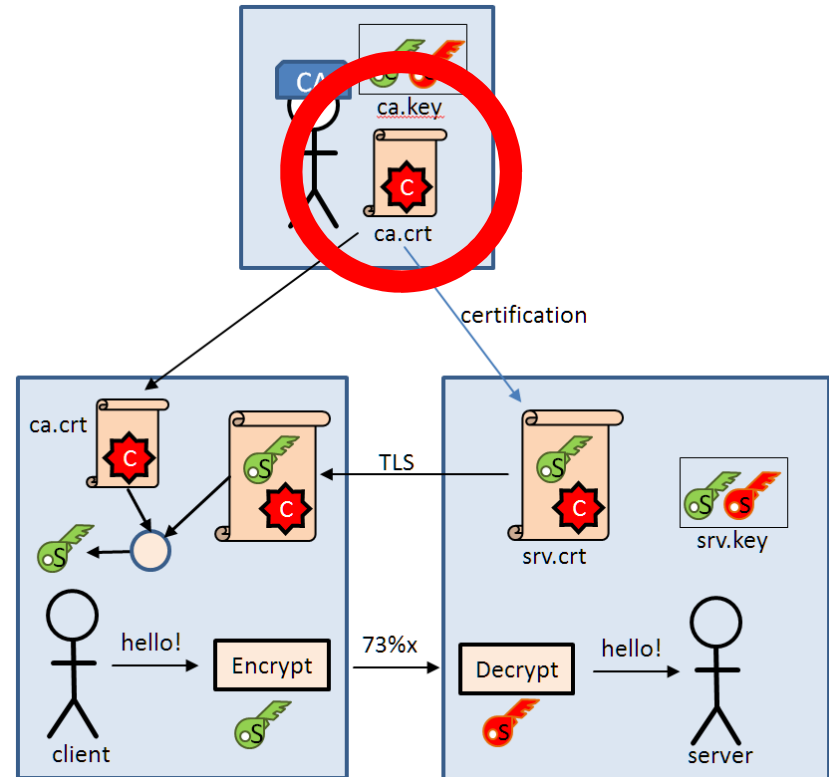
My work (and challenges) with PKI

- 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



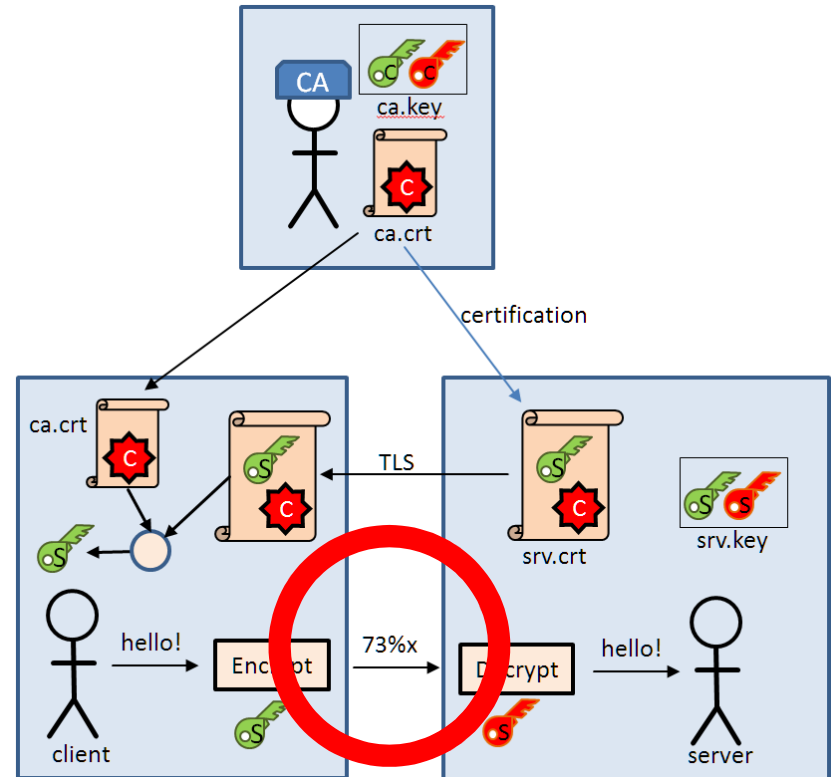
My work (and challenges) with PKI

- 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



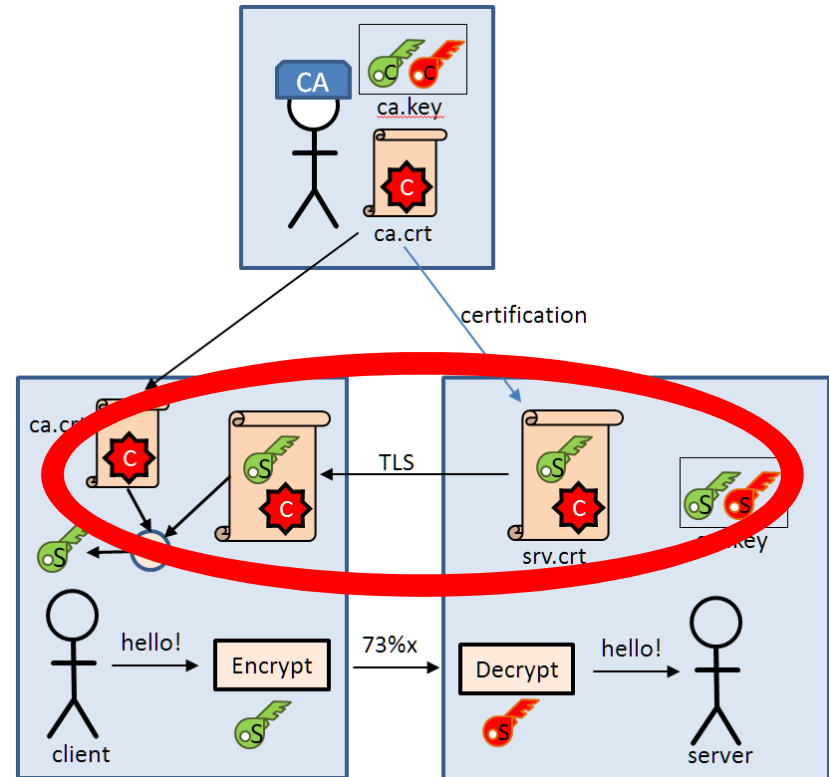
My work (and challenges) with PKI

- 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

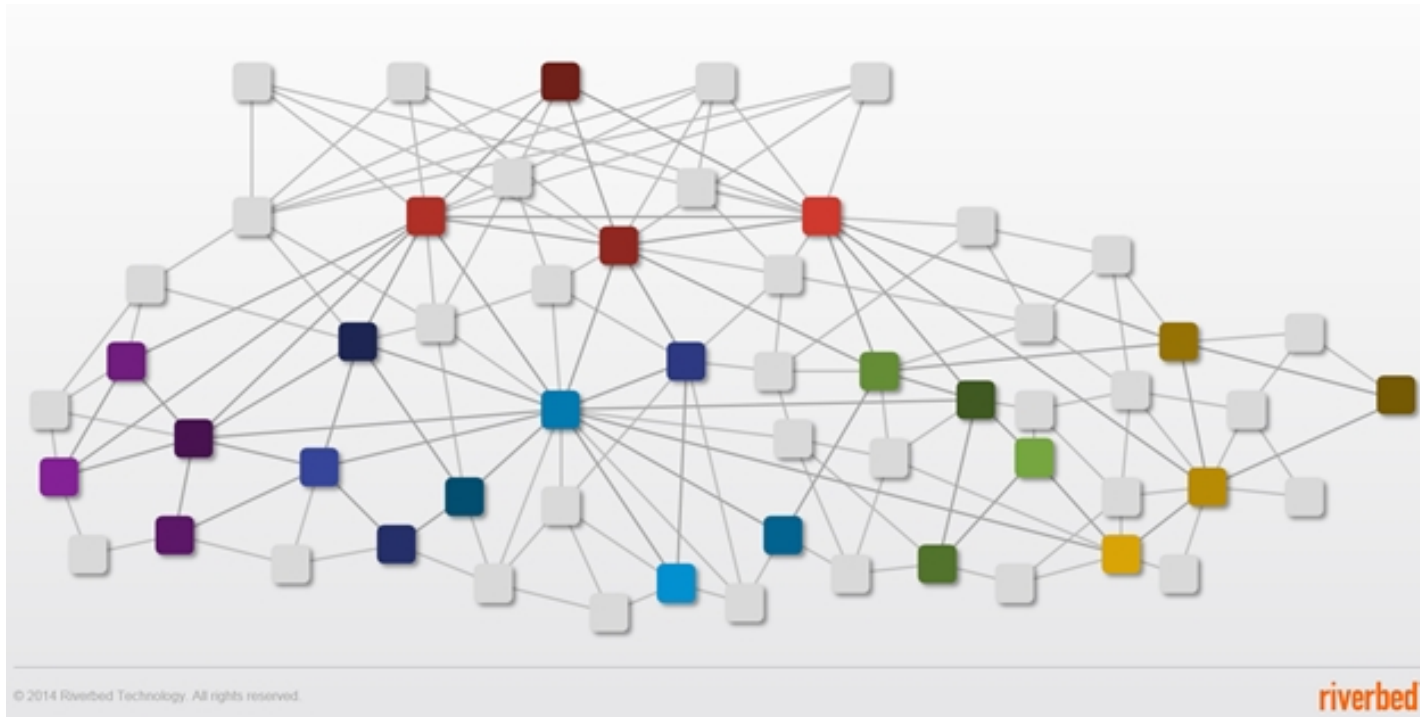
```
openssl x509 -inform der -in certificatename.der -out  
certificatename.pem
```



The screenshot shows a Stack Overflow page for the question "How to create a certificate with keytool?". The page includes a search bar, a navigation menu with "Stack Overflow" selected, and a "Teams" section. The main content area shows the question title and two answers. The first answer, by user "hitta.se", states: "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." The second answer, by user "cacerts", states: "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?" The question is tagged with "java", "https", "glassfish", and "keytool".

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?

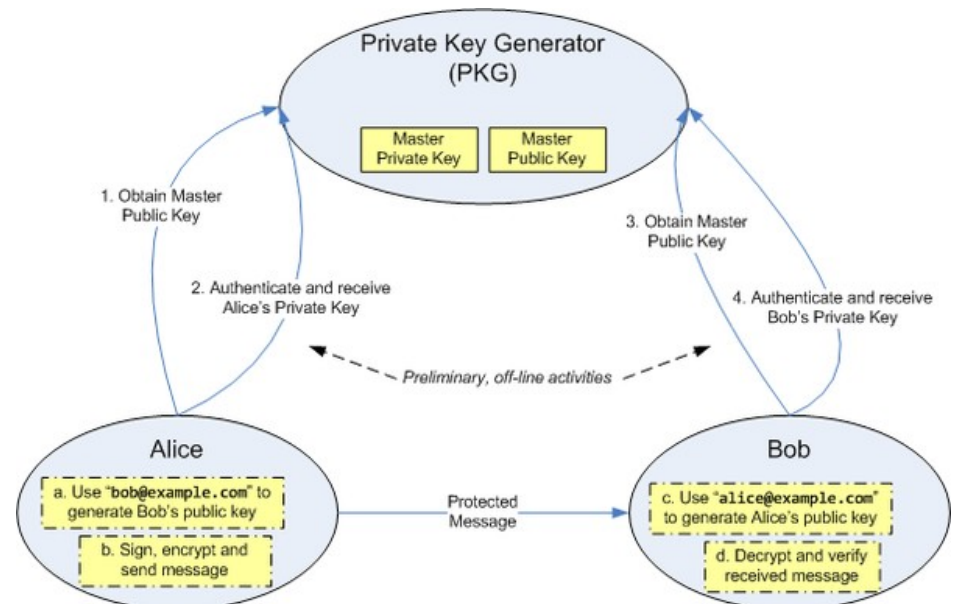


Alternatives to traditional PKI

- “...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.



[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

