PKI IN THE REAL WORLD

HÅVARD RADDUM

SIMULA@UIB

Last changed 10.05.2017

OVERVIEW

- How root certificates get into browsers
- Hacked CAs
- Doubtful PKI practices
- Efforts to gain trust in certificates
- Lessons learned from real-world PKI

HOW DO ROOT CERTIFICATES GET INTO OPERATING SYSTEMS?

MICROSOFT TRUSTED ROOT PROGRAM

- Microsoft runs a trusted root program where CAs may get their root certificates included in Windows OS
- A new CA who wants their root certificate included in Windows must submit:
    - An application with the CA's physical address and two contact persons
    - Report from an approved auditor

MICROSOFT TRUSTED ROOT PROGRAM

- Distinguishes between commercial CAs and government CAs
- Government CA may audit itself, but only allowed to issue certificates to domains belonging to the same country
- Commercial CA must be audited annually by an auditor from list given by Microsoft
  - Auditors in 25 countries, companies like Ernst & Young, Deloitte Touche, KPMG and PwC heavily represented
- Apart from cost of audit, becoming a Microsoft-approved root CA is free

APPLE ROOT CERTIFICATE PROGRAM

- Apple runs root certificate program for including root certificates in OS X and iOS
- CA who wants their root certificate included in OS X/iOS must:
  - Send email to Apple with contact names, company details and explanation of how the certificate will benefit Apple customers
  - Engage an auditor to do a WebTrust audit of the CA's business

APPLE ROOT CERTIFICATE AUDIT

- Apple does not require audit to come from list of pre-approved WebTrust auditors
- If auditor is unknown to WebTrust, CA must (somehow) prove the audit is equivalent to the WebTrust audit
- No requirement to periodically renew audit
- Apple does not charge any payment for including root certificates in OS X/iOS

OTHER PLATFORMS

- Mozilla has similar policy for including root certificates in Mozilla products (Firefox and Thunderbird):
  - Contact persons and company details of CA
  - Audit report from ETSI, WebTrust or similarly approved auditor
  - Free of charge to get root certificate in Mozilla
- Android: Submit request to include root certificate through Google's bug tracker for Android

HACKED CA'S

DIGINOTAR

- DigiNotar was a Dutch CA owned by VASCO
- DigiNotar's root certificate included in all browsers at the time they were attacked (June 2011 or earlier)
- Fake certificates for *.google.com had been signed by DigiNotar's private key
- Became clear that DigiNotar's CA system had been hacked
- Attackers used fake certificates for MitM attack on Iranian gmail users

DIGINOTAR

- Fake certificates for several other domains signed by DigiNotar have also appeared:
  - *.microsoft.com
  - *.wordpress.com
  - *.windowsupdate.com
- Google and Mozilla were first to remove DigiNotar's root certificate from Chrome and Firefox, other browsers followed
- DigiNotar went bankrupt on September 20th 2011

COMODO

- Largest CA on the internet - 40-50% market share*
- Company originated in the UK, now based in USA
- March 2011 - user account of one of Comodo’s RAs was compromised
- Attacker successfully made Comodo sign 9 fake certificates
- Attack quickly discovered, fake certificates immediately revoked

* http://www.whichssl.com/compare-ssl-certificates.html

* https://w3techs.com/technologies/overview/ssl_certificate/all

COMODO

- RA was suspended from Comodo’s operations
- Trust in Comodo not reduced to the extent that root certificates were removed from browsers
- Revoking Comodo’s root certificates would at the time affect 85.000 - 200.000 different web sites
- Would cause major problems for sites and users if all certificates coming from Comodo failed to validate
- Is Comodo too big to fail?

TOO BIG TO FAIL

- Browser vendors face a very difficult choice if trust in a big CA is compromised
- Revoke CA’s root certificate(s)
  - Will cause browser to give alarm to users when trying to set up TLS-connection to web site with certificate under big CA
  - Major problem for users and web sites
- Not revoke CA’s root certificate(s)
  - Users vulnerable to man-in-the-middle attacks

US GOVERNMENT INTERFERENCE?

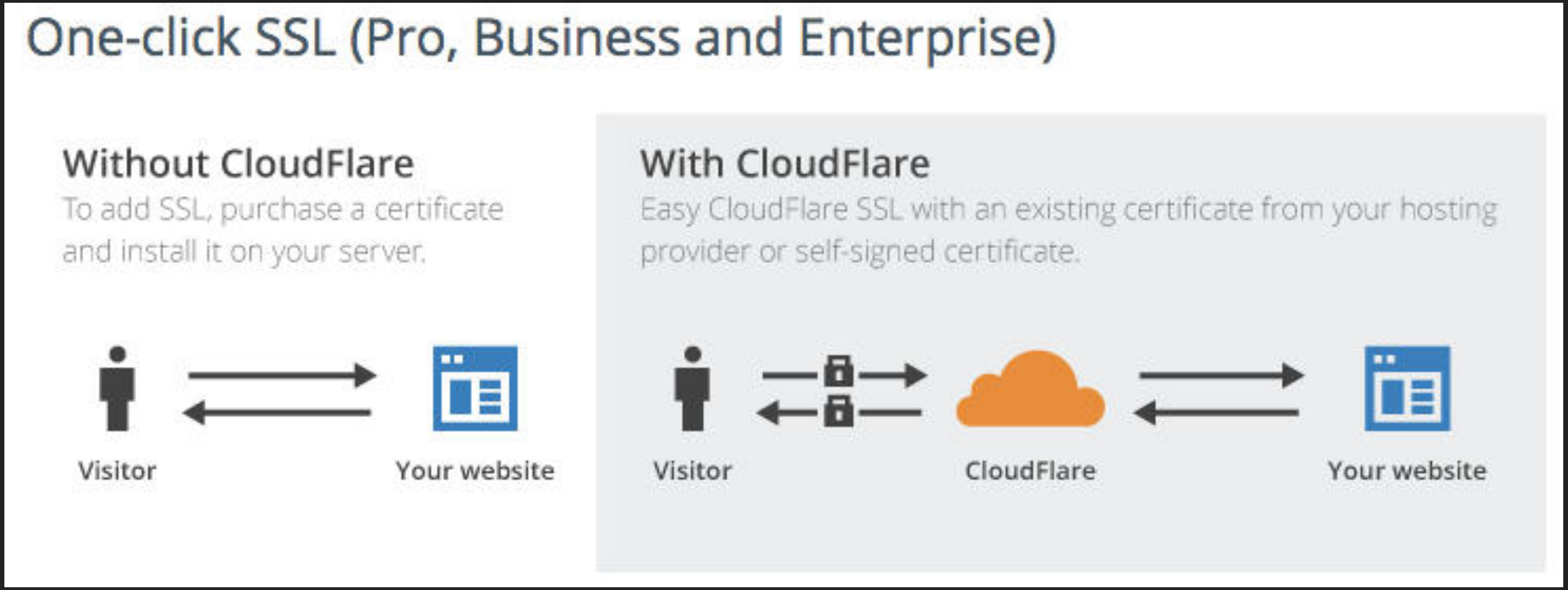
- Snowden disclosures has shown FISC serve court orders to IT- and phone companies:
  - Demand privileged access to data for NSA
  - Demand the fact of such access be kept secret
- May speculate that American CAs have been served similar orders to hand over private key for root certificates
- Case of Lavabit is example of such order issued to holder of web server certificate

DOUBTFUL PKI PRACTICES

CLOUDFLARE

- CloudFlare is a company based in USA selling services in web-hosting
- One of their products is «one-click SSL», or «flexible SSL»
  - Web server www.example.com using «flexible SSL» do not need to get a certificate from a CA
  - Do not need to enable TLS on www.example.com, do not need to have a private key
  - Browser of visitors to www.example.com will still show the connection to be valid TLS!

CLOUDFLARE'S SSL AS A SERVICE (2014)



One-click SSL (Pro, Business and Enterprise)

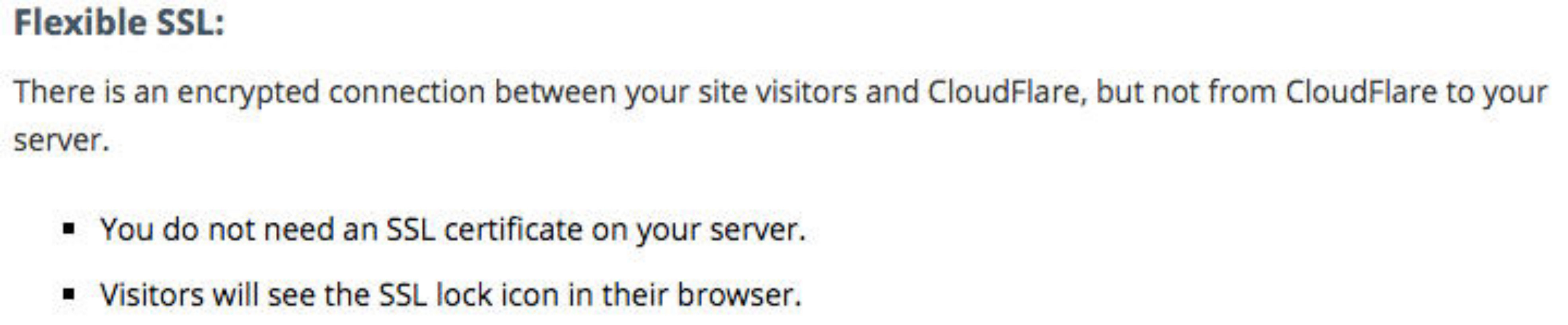
Without CloudFlare

To add SSL, purchase a certificate and install it on your server.

With CloudFlare

Easy CloudFlare SSL with an existing certificate from your hosting provider or self-signed certificate.

www.cloudflare.com



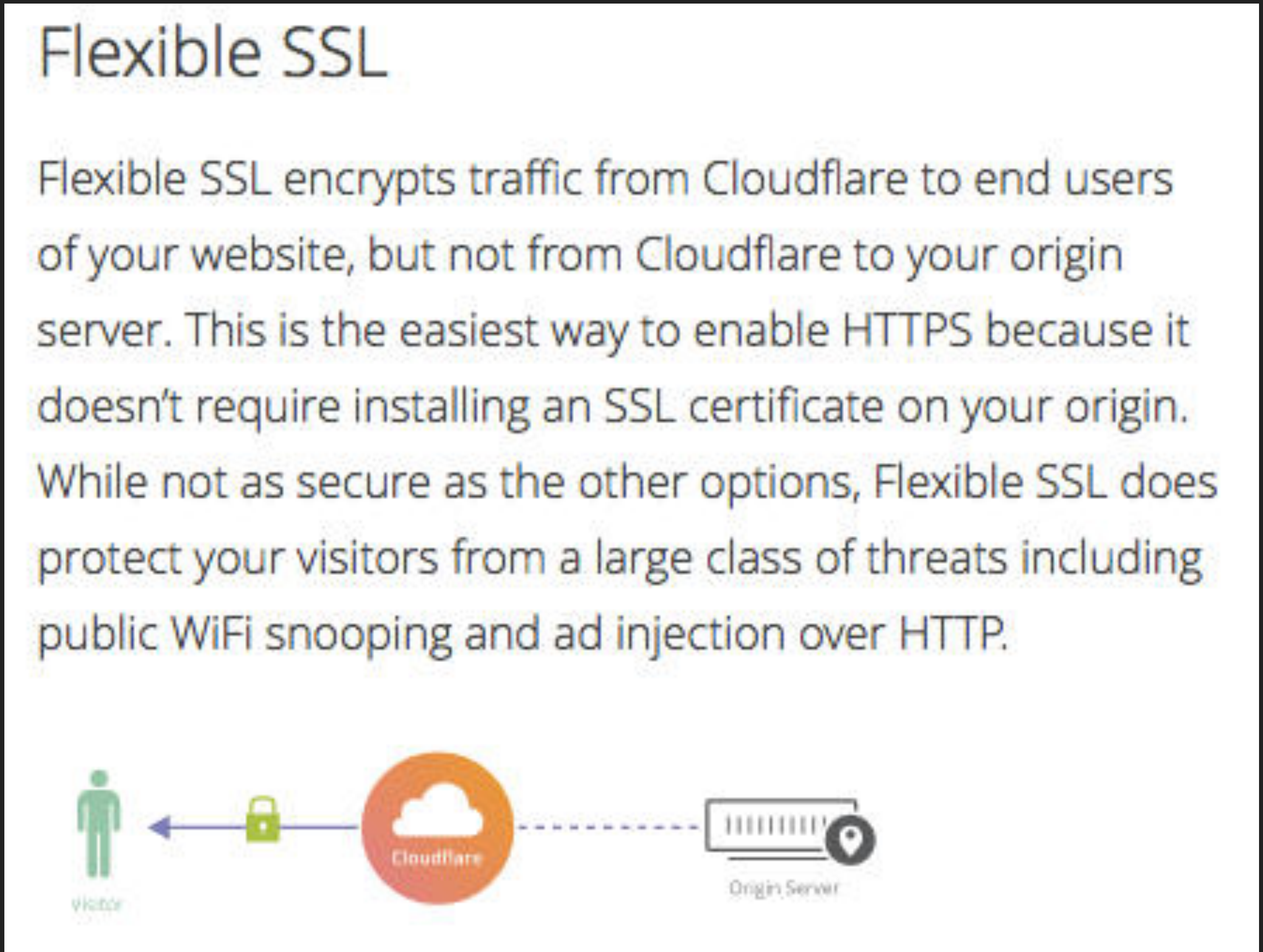
Flexible SSL:

There is an encrypted connection between your site visitors and CloudFlare, but not from CloudFlare to your server.

- You do not need an SSL certificate on your server.
- Visitors will see the SSL lock icon in their browser.

www.cloudflare.com

CLOUDFLARE'S SSL AS A SERVICE (2016)



Flexible SSL

Flexible SSL encrypts traffic from Cloudflare to end users of your website, but not from Cloudflare to your origin server. This is the easiest way to enable HTTPS because it doesn't require installing an SSL certificate on your origin. While not as secure as the other options, Flexible SSL does protect your visitors from a large class of threats including public WiFi snooping and ad injection over HTTP.

www.cloudflare.com

HOW DOES FLEXIBLE SSL WORK?

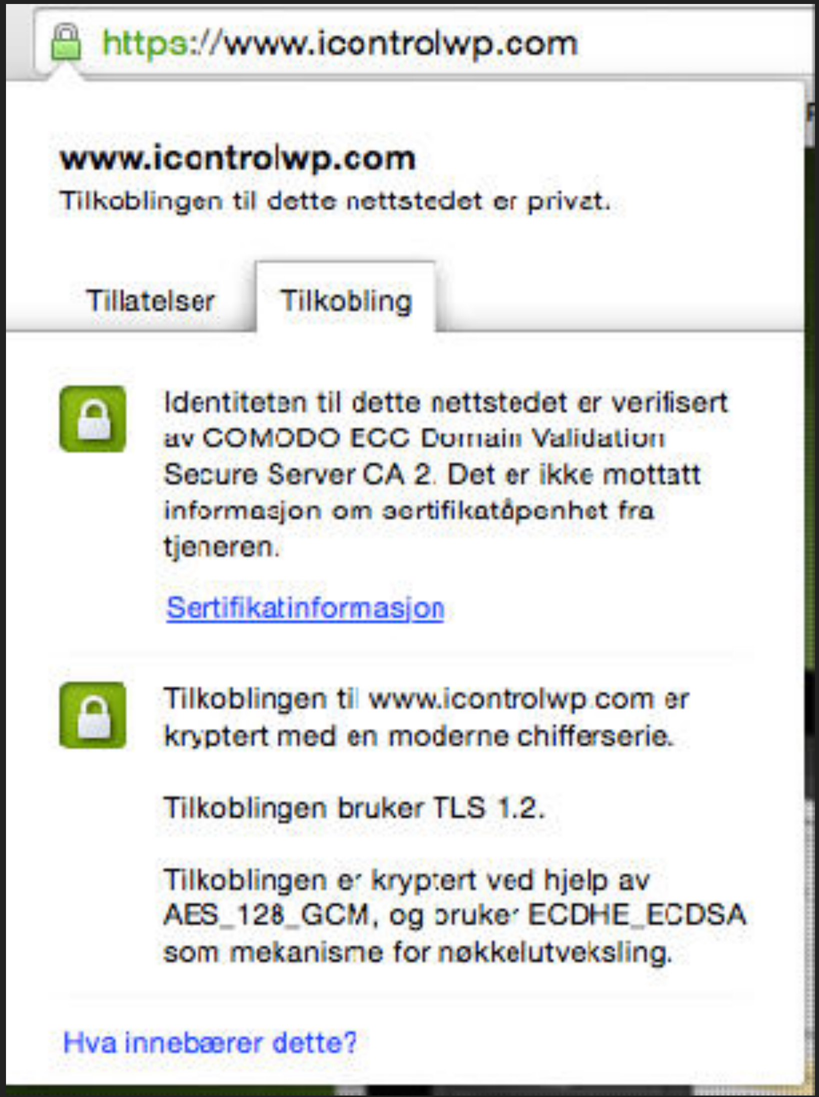
- Visitors to www.example.com have a valid TLS-connection to CloudFlare, even though address bar in browser says www.example.com

- CloudFlare decrypts visitor’s traffic and sends it in plaintext to www.example.com

- Certificate CloudFlare is using is issued to CloudFlare, but has www.example.com as an «alternative name» in Subject Alt Name extension

FLEXIBLE SSL EXAMPLE

- https://www.icontrolwp.com/ is using CloudFlare’s flexible SSL certificate
- Visit site and examine certificate



https://www.icontrolwp.com

www.icontrolwp.com
Tilkoblingen til dette nettstedet er privat.

Tillatelser | Tilkobling

Identiteten til dette nettstedet er verifisert av COMODO ECC Domain Validation Secure Server CA 2. Det er ikke mottatt informasjon om sertifikatåpenhet fra tjeneren.

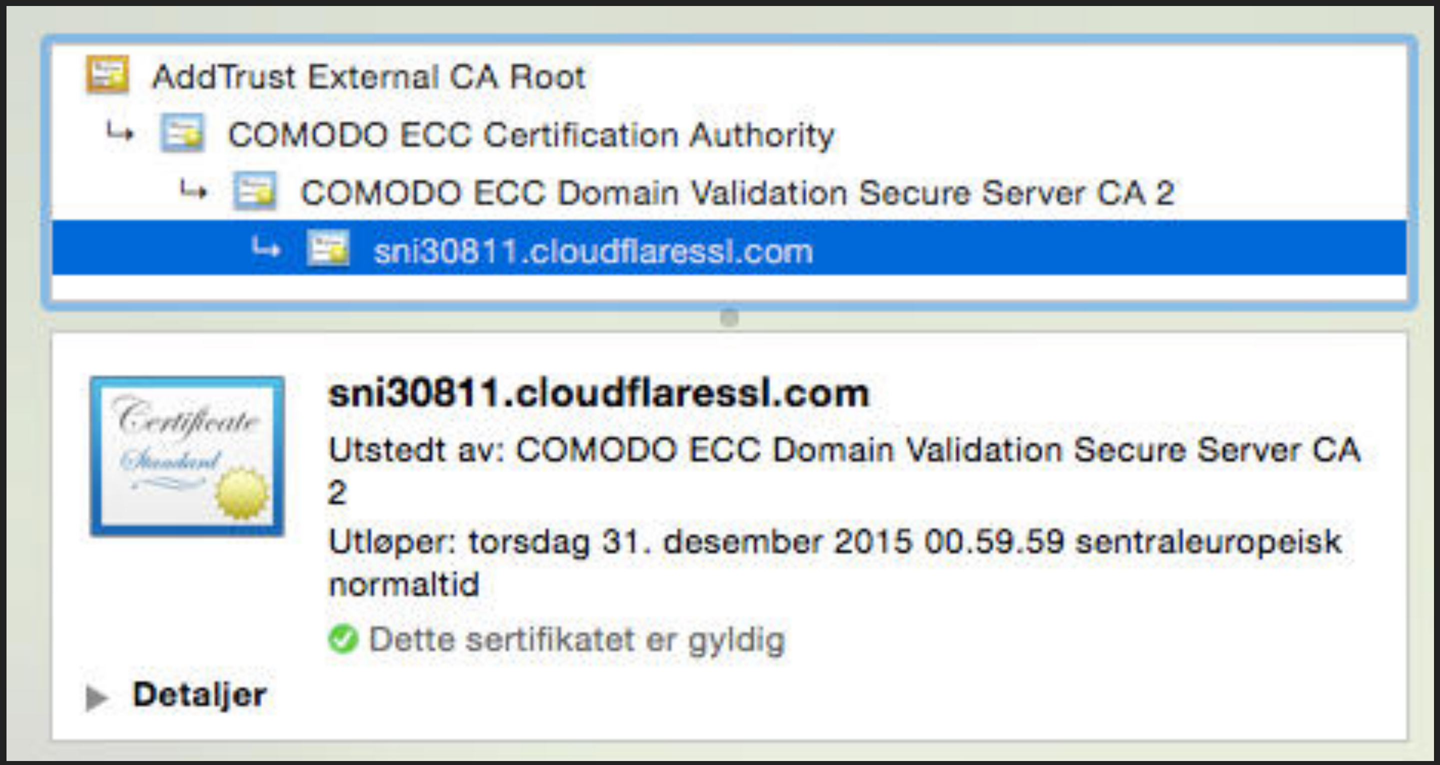
Sertifikatinformasjon

Tilkoblingen til www.icontrolwp.com er kryptert med en moderne chifferserie.

Tilkoblingen bruker TLS 1.2.

Tilkoblingen er kryptert ved hjelp av AES_128_GCM, og bruker ECDHE_ECDSA som mekanisme for nøkkelutveksling.

Hva innebærer dette?



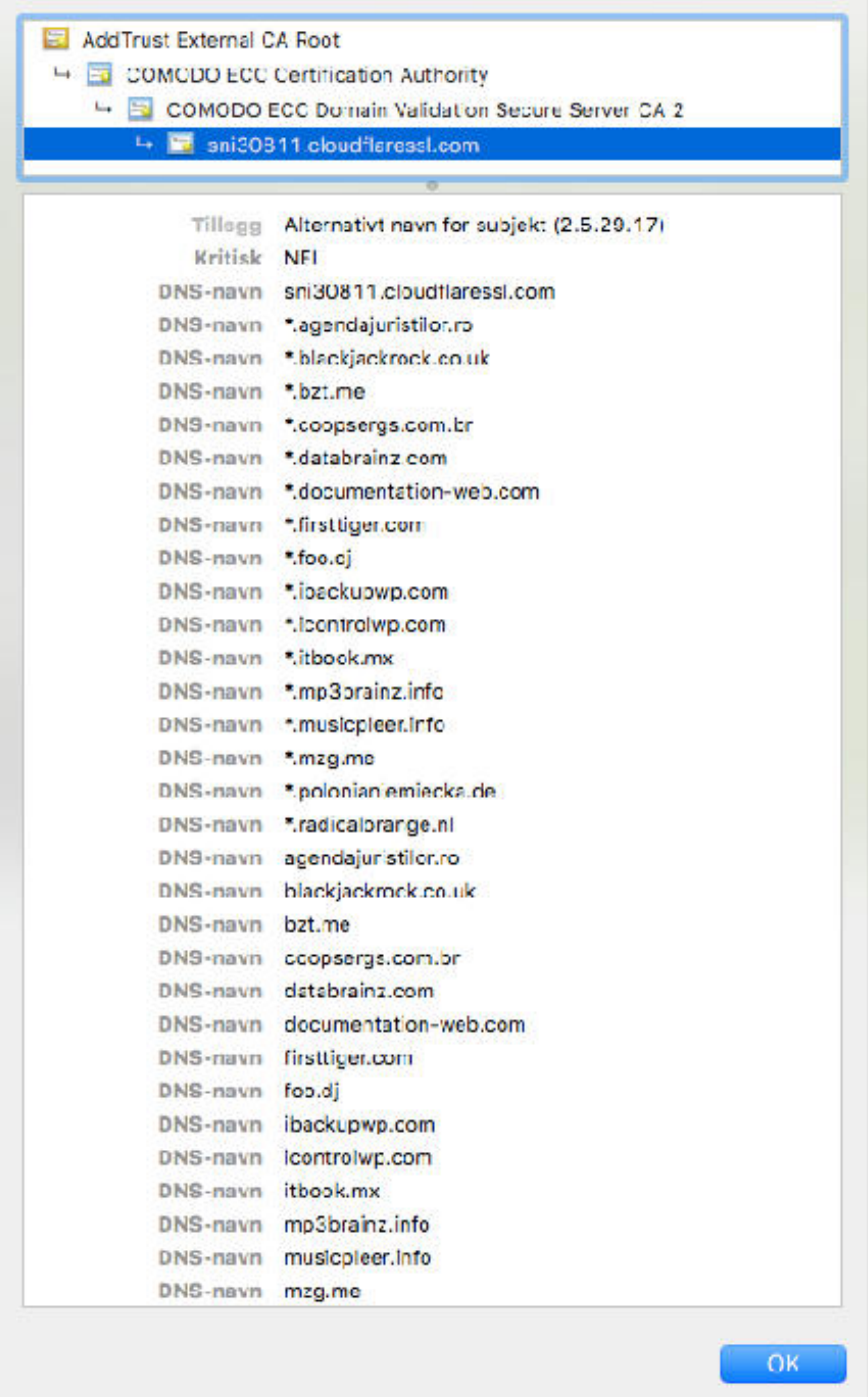
AddTrust External CA Root
↳ COMODO ECC Certification Authority
↳ COMODO ECC Domain Validation Secure Server CA 2
↳ sni30811.cloudflaressl.com

sni30811.cloudflaressl.com
Utstedt av: COMODO ECC Domain Validation Secure Server CA 2
Utløper: torsdag 31. desember 2015 00.59.59 sentraleuropeisk normaltid
✓ Dette sertifikatet er gyldig

▶ **Detaljer**

ALT NAME EXTENSION

- CloudFlare's certificate can be used for any of these domains



AddTrust External CA Root
↳ COMODO ECC Certification Authority
↳ COMODO ECC Domain Validation Secure Server CA 2
↳ sni30811.cloudflaressl.com

Tillegg	Alternativt navn for subjekt (2.5.29.17)
Kritisk	NEI
DNS-navn	sni30811.cloudflaressl.com
DNS-navn	*.agendajuristilor.ro
DNS-navn	*.blackjackrock.co.uk
DNS-navn	*.bzt.me
DNS-navn	*.coopsergs.com.br
DNS-navn	*.databrainz.com
DNS-navn	*.documentation-web.com
DNS-navn	*.firsttiger.com
DNS-navn	*.foo.dj
DNS-navn	*.ibackupwp.com
DNS-navn	*.icontrolwp.com
DNS-navn	*.itbook.mx
DNS-navn	*.mp3brainz.info
DNS-navn	*.musicpleer.info
DNS-navn	*.mzg.me
DNS-navn	*.polonianiemiecka.de
DNS-navn	*.radicalorange.nl
DNS-navn	agendajuristilor.ro
DNS-navn	blackjackrock.co.uk
DNS-navn	bzt.me
DNS-navn	coopsergs.com.br
DNS-navn	databrainz.com
DNS-navn	documentation-web.com
DNS-navn	firsttiger.com
DNS-navn	foo.dj
DNS-navn	ibackupwp.com
DNS-navn	icontrolwp.com
DNS-navn	itbook.mx
DNS-navn	mp3brainz.info
DNS-navn	musicpleer.info
DNS-navn	mzg.me

CA ISSUING FLEXIBLE SSL CERTIFICATES

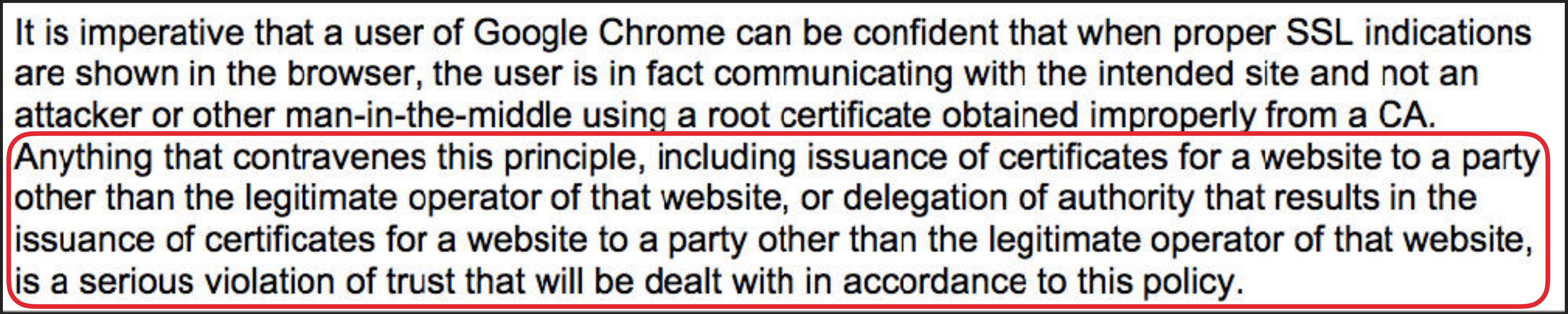
- Comodo is the CA issuing CloudFlare's flexible SSL certificates
- Flexible SSL certificate issued to CloudFlare has a long list of domains in Subject Alt Name extension
- Subject Alt Name extension supposed to contain aliases and specific domains for the certificate owner
- Comodo is signing on information it knows is not true, or Subject Alt Name extension has lost its meaning

FLEXIBLE SSL PRO AND CON

- Supporters of Flexible SSL:
  - A web site not buying a certificate and configuring their server for TLS will only be able to communicate unencrypted
  - Flexible SSL is an improvement, because traffic is encrypted at least part of the way
- On the other hand:
  - The user is being lied to when Flexible SSL is in use

VIOLATION OF GOOGLE'S CA POLICY?

- Excerpt from Google's root certificate policy:



It is imperative that a user of Google Chrome can be confident that when proper SSL indications are shown in the browser, the user is in fact communicating with the intended site and not an attacker or other man-in-the-middle using a root certificate obtained improperly from a CA. Anything that contravenes this principle, including issuance of certificates for a website to a party other than the legitimate operator of that website, or delegation of authority that results in the issuance of certificates for a website to a party other than the legitimate operator of that website, is a serious violation of trust that will be dealt with in accordance to this policy.

http://www.chromium.org/Home/chromium-security/root-ca-policy

- So why does Google Chrome accept CloudFlare's and Comodo's practice?

- Is Comodo too big to fail?

WOSIGN AND STARTCOM

- WoSign was a Chinese CA and StartCom was an Israeli CA
- In November 2015 WoSign acquired StartCom
- When one CA buys another it must be publicly disclosed to the CA/Browser forum
- For unknown reasons, WoSign has tried to keep the acquisition secret and has argued StartCom continues to operate as an independent CA
- Team from Mozilla has gathered lots of evidence that StartCom is using (a copy of) WoSign's infrastructure for certificate creation

SHA-1

- SHA-1 is a hash algorithm, can be used to generate digital signatures for certificates
- The security of SHA-1 has deteriorated to a level where the CA/Browser forum wants to phase out SHA-1
- Decided that certificates issued in 2016 and later can not use SHA-1 for digital signing - browsers will object
- May be costly for customers of CAs to upgrade their software to support accepted hash algorithms

WOSIGN AVOIDING SHA-1 BAN

- Discovered in September 2016 that WoSign has issued certificates in 2016 using SHA-1
- Avoid triggering browser alarm by back-dating validity period to start in late 2015
- Evidence these certificates were manually modified and not automatically generated by WoSign's system

RESPONSE

- Team from Mozilla lead investigation on WoSign's practice
- Mozilla products started to distrust WoSign/StartCom certificates on October 21st 2016
- Mozilla also no longer accepts audits from Ernst & Young's Hong Kong office.
- Apple announced in October 2016 they would also distrust WoSign/StartCom certificates

GOOGLE VS SYMANTEC

- Symantec is a big computer security firm headquartered in Silicon Valley and has more than 21.000 employees
- Symantec owns and operates several CAs:
  - GeoTrust, Thawte, CrossCert, Certisure,…
- Symantec is (arguably) second or third largest operator of root CAs

GOOGLE VS SYMANTEC

- In October 2015, Google noticed Symantec had issued certificates for www.google.com and google.com without Google’s knowledge
- Initial investigation by Google found 127 mis-issued certificates
- Symantec acknowledged the fact and fired persons responsible for issuing the certificates
- However, Symantec also emphasized that no harm was done and that the certificates were only made for testing purposes

GOOGLE VS SYMANTEC

- Google continued investigation, reported in March 2017 that 30.000+ certificates had been mis-issued over several years
- Google faults Symantec for lax policies and controls regarding issuance of certificates
- Four of Symantec's subsidiaries were responsible for the 30.000+ mis-issued certificates
- Symantec claims Google grossly overstates the problem

GOOGLE VS SYMANTEC

- Google announced in March 2017 that Google Chrome would start to gradually distrust certificates issued by CAs owned by Symantec
  - Chrome 59 will only accept Symantec certificates with validity period of 33 months or less
  - Accepted validity period decreases by 6 months for each Chrome release
  - Chrome 64 will only accept certificates valid for < 9 months

GOOGLE VS SYMANTEC

- Google will also treat any Symantec EV certificates as «ordinary» certificates for at least one year
- Google explains that the gradual distrust of Symantec is introduced not to disrupt too many users and services on the internet, acknowledging that Symantec is too big to fail
- Symantec calls Google's decision for «unexpected and irresponsible»

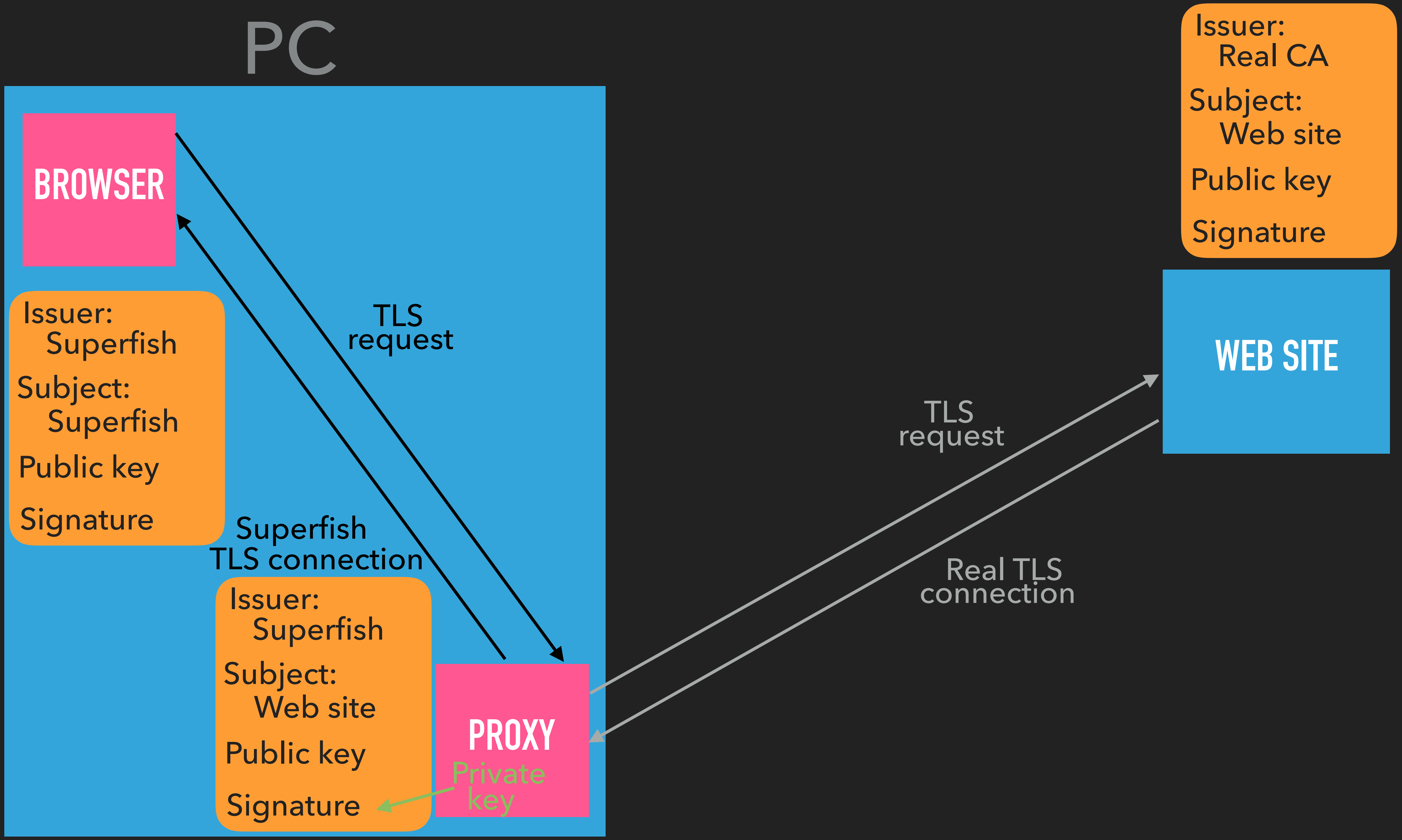
LENOVO, SUPERFISH AND KOMODIA

- Lenovo is a PC manufacturer from China
- In the period August 2014 - January 2015 Lenovo sold Windows laptops with pre-installed root certificate from Superfish
- Laptops also had software that would include a proxy on the laptop as a man-in-the-middle in any TLS-connection

HOW IT WORKS

- A Superfish-infected laptop contains a proxy intercepting all web traffic between browser and website
- Proxy also acts as a local CA
- When browser wants to set up TLS to some web site:
  - Regular TLS-connection set up between proxy and web site
  - Proxy issues certificate on the fly for web site, signed by Superfish certificate
  - Proxy sets up TLS-connection between itself and browser, using the just-issued certificate

HOW IT WORKS – MAN IN THE MIDDLE



WHAT IS THE SECURITY PROBLEM?

- Anyone knowing the private key for the Superfish root certificate can issue certificates for any web site
- These certificates will be accepted as valid by all Superfish-infected laptops
- Attacker may become man-in-the-middle in a TLS connection to any web site, if he knows private key for Superfish root certificate

SUPERFISH PRIVATE KEY

- Private key resides in the proxy on every Superfish-infected laptop
- Private key protected by password
- One analyst found the private key, working only for three hours, password was ‘komodia’
- Komodia is name of company developing the TLS-proxy
- The same private key used on all infected laptops
- Private key for forging certificates is public knowledge!

WHY DID THEY DO IT?

- Superfish dynamically adds advertisements to web pages
- Problem for Superfish: How to add advertisement in web page secured by TLS?
- Solution: Set up proxy/local CA on each machine
  - Decrypt TLS-protected web page
  - Add the advertisement
  - Re-encrypt web page for the proxy - browser link

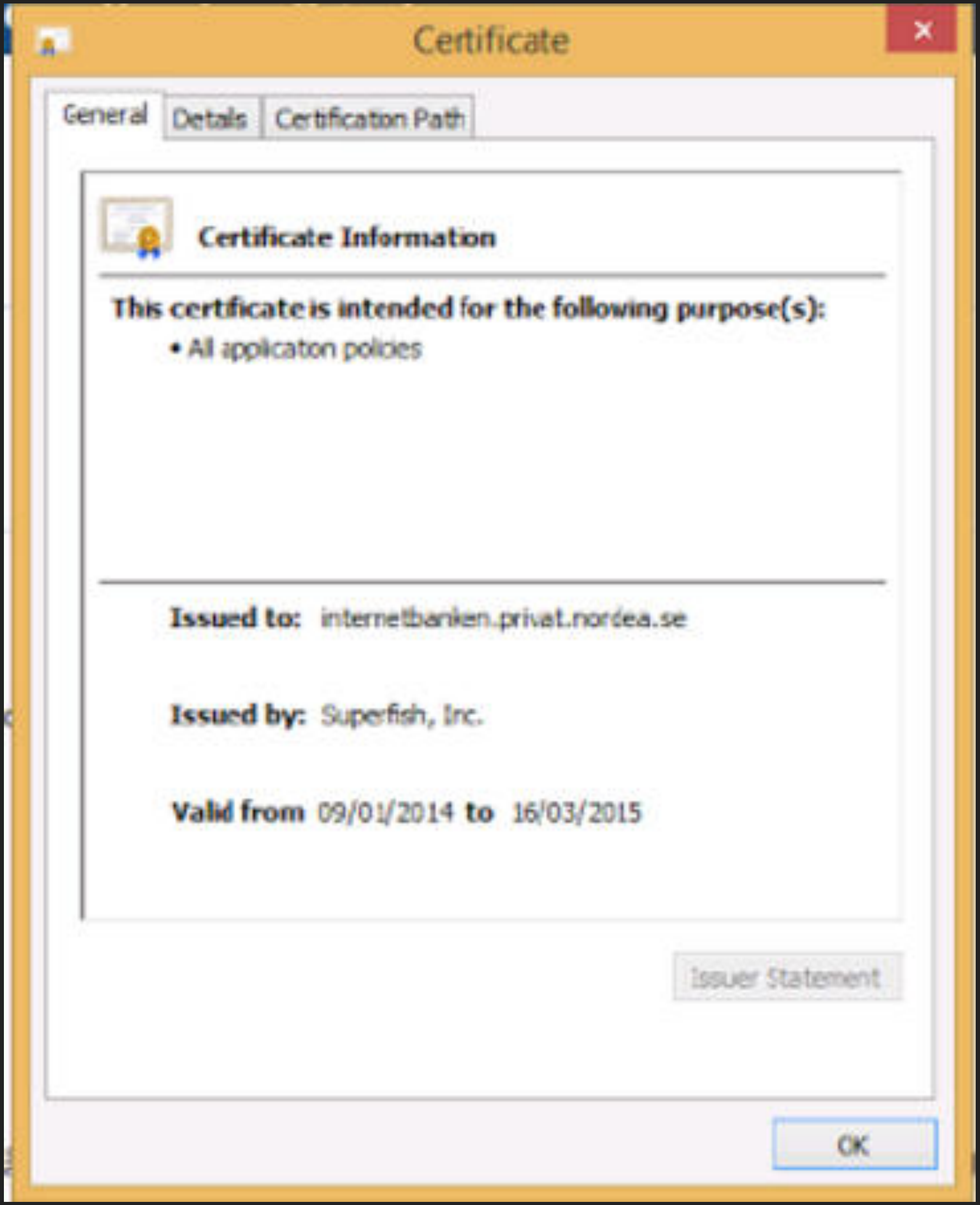
WHY DID LENOVO PLAY ALONG?

- New problem for Superfish: How to get Superfish root certificate into OS of laptops?
  - Adding root certificates requires privileged access
- Solution: Have root certificates (and proxy) pre-installed from PC manufacturer
- Estimated that Lenovo made approximately $250.000 by agreeing to produce Superfish-infected laptops

LENOVO AND SUPERFISH RESPONSE

- Both Lenovo and Superfish claimed initially there was no security problem with their practice when it made headlines
- After attack scenarios were presented in detail:
    - Lenovo cut off all cooperation with Superfish, and started helping customers remove proxy and root certificate from infected machines
    - Superfish blamed Komodia for «inadvertently introducing the vulnerability»
    - Komodia did not make any comments

PKI IN THE REAL WORLD



Certificate Information

This certificate is intended for the following purpose(s):

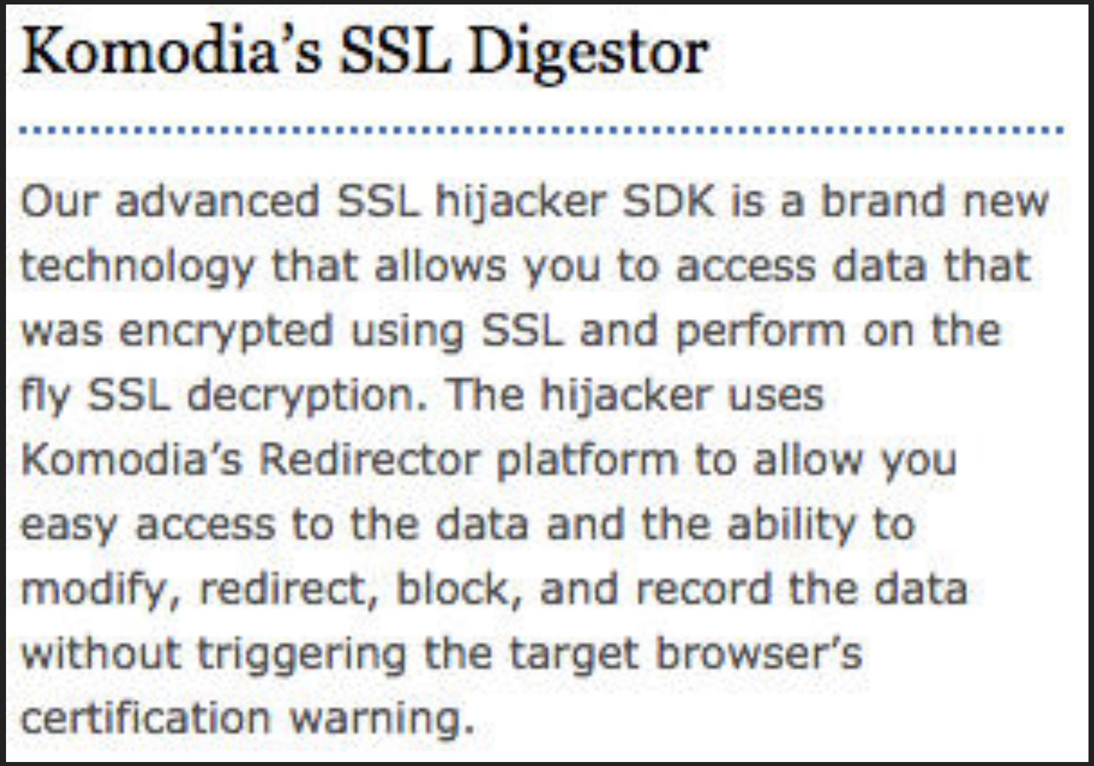
- All application policies

Issued to: internetbanken.privat.nordea.se

Issued by: Superfish, Inc.

Valid from 09/01/2014 **to** 16/03/2015

- Komodia is an Israeli company
- Front page of komodia.com:



Komodia's SSL Digestor

Our advanced SSL hijacker SDK is a brand new technology that allows you to access data that was encrypted using SSL and perform on the fly SSL decryption. The hijacker uses Komodia's Redirector platform to allow you easy access to the data and the ability to modify, redirect, block, and record the data without triggering the target browser's certification warning.

www.komodia.com

EFFORTS TO GAIN TRUST IN CERTIFICATES AND PKI

LACK OF TRUST

- The PKI model assumes there exist CAs that users trust
- Incidents we have seen erode the trust in CAs and PKI
- Two efforts to rebuild trust in PKI:
  - Extended Validation
  - Certificate pinning

EXTENDED VALIDATION

- Work initiated by Comodo in 2005
- First standard of Extended Validation (EV) in June 2007
- Problem EV tries to solve:
  - CAs are supposed to check identity of subjects they issue certificates to
  - Over the years the practice of ID-check has deteriorated
  - CAs issue certificates to domains, where a simple URL domain name is the only identifier of a subject

EXTENDED VALIDATION

- EV certificates require:
  - A physical address for the subject requesting certificate
  - An organisation number or similar for the subject
  - Contact person responsible for the certificate request
  - CA must verify the submitted information from independent sources
  - CA must make phone calls to contact person and company

EXTENDED VALIDATION

- Only CAs who has been audited by a WebTrust auditor may issue EV certificates
- Criticism of EV:
  - Only solves the authentication problem the CA should solve in the first place (without EV)
  - EV certificates are (much) more expensive than regular certificates, introduced to boost earnings of CAs
  - EV doesn’t make much difference in practice, vast majority of users don’t understand the security of certificates anyway

CERTIFICATE PINNING

- PKI model assumes the user makes a choice of which CAs to trust
- In reality, users makes no choice but implicitly «trusts» the root CAs included in their operating system or browser
- Problem with long lists of implicitly trusted root CAs:
  - Any of the 100+ CAs can issue valid certificates for any company/organisation/domain
  - If only one CA is compromised, attacker can gain MitM access on TLS-connection from any user to any web server

CERTIFICATE PINNING

- Certificate pinning attaches particular certificates or public keys to the browser (or other application) for making TLS-connections
- Certificate pinning may be achieved by
  - Hardcoding a particular certificate into the application
  - Storing certificate on client the first time a user makes TLS-connection to a particular web server

CERTIFICATE PINNING

- Client does not need to store whole certificate, only hash value of pinned certificate is needed to verify a received certificate is the same as expected
- For successful MitM attack on client with certificate pinning:
  - Certificate not hardcoded in application
  - Fake certificate presented to client on first use
  - Fake certificate presented to client on all future uses

CERTIFICATE PINNING

- Google Chrome contains hardcoded hash of genuine certificate for *.google.com
- Mechanism detected the fake certificate from DigiNotar
- Certificate pinning reduces the need for complete and blind trust in all existing CAs

LESSONS LEARNED FROM REAL-WORLD PKI

UNEDUCATED USERS

- PKI model assumes users trust CAs, but:
  - Most users don't know how PKI works
  - Are not aware of implicit trust in long list of root CAs
  - Do not understand exactly what assurances TLS gives
  - Do not understand certificates
  - Do not know what to look for if clicking to examine certificate
- In practice, browser vendors have taken on the responsibility of verifying which CAs to trust

BUSINESS MODELS VS SECURITY MODELS

- CAs are more businesses than authorities
- Business models are sometimes contrary to security models
  - Buyers of certificates present «creative» solutions to some problem involving certificates
  - CAs listen and may be willing to stretch their own policies
- CAs may become so big, that removing their trust will cause major practical problems

BOTTOM LINE

- Security models like PKI implicitly assumes that all parties understand the model
- This is not true in the real world
- Practices involving certificates that should be reacted upon goes unnoticed
- Certificate pinning may be a first step towards a new trust model, relying less on CAs

SOURCES

- https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html
- http://www.vasco.com/company/press_room/news_archive/2011/news_vasco_announces_bankruptcy_filing_by_diginotar_bv.aspx
- http://blog.gerv.net/2011/09/updated-diginotar-cn-list/
- https://www.eff.org/deeplinks/2011/03/iranian-hackers-obtain-fraudulent-https
- https://www.eff.org/files/ccc2010.pdf
- http://www.chromium.org/Home/chromium-security/root-ca-policy
- http://arstechnica.com/security/2015/02/lenovo-pcs-ship-with-man-in-the-middle-adware-that-breaks-https-connections/
- http://www.forbes.com/sites/thomasbrewster/2015/02/27/lenovo-got-very-little-from-superfish-deal/
- http://social.technet.microsoft.com/wiki/contents/articles/31633.microsoft-trusted-root-program-requirements.aspx
- https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/policy/inclusion/

X.509 certificates and the PKI model

Håvard Raddum
Simula@UiB

Outline

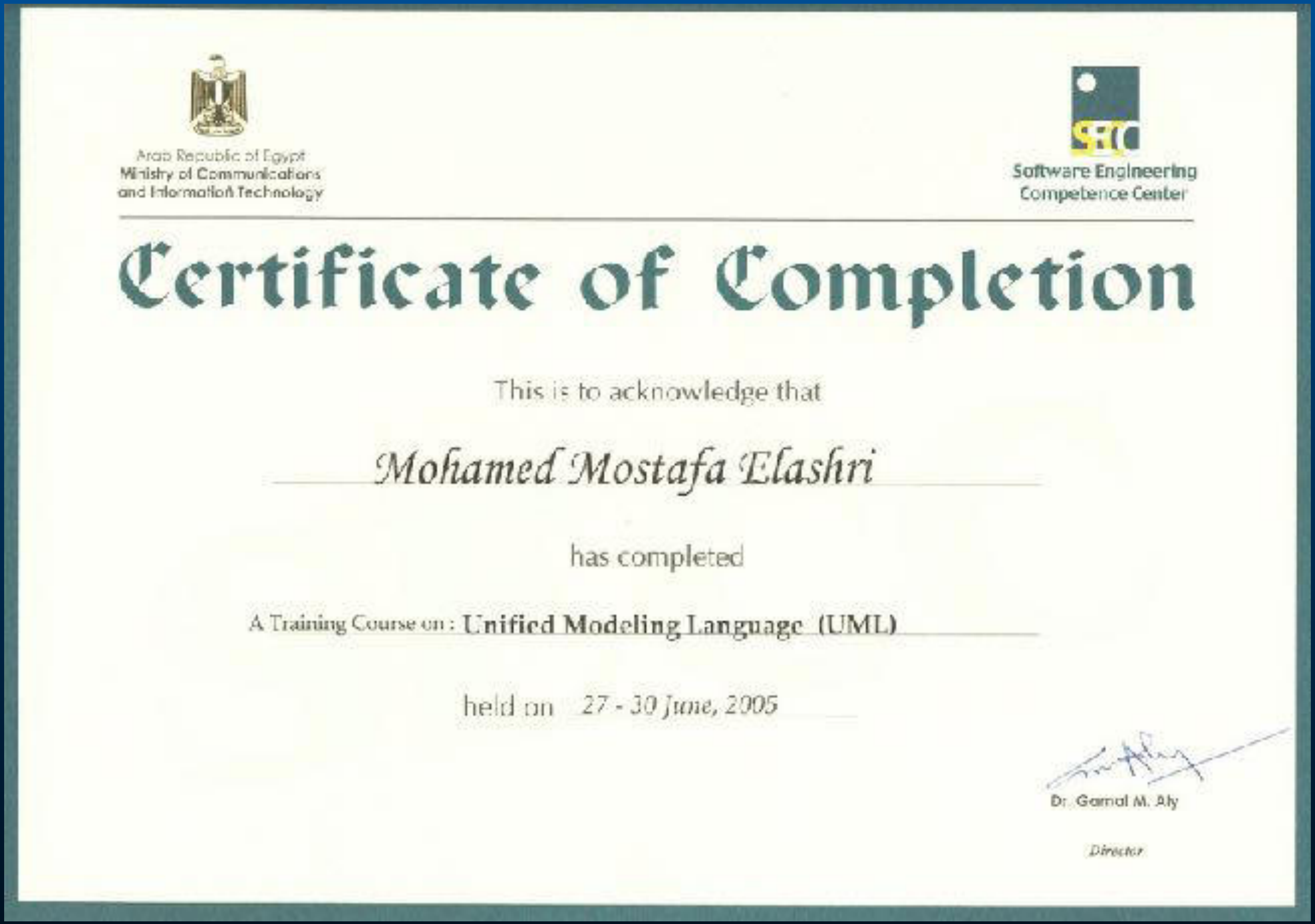
- Certificates and trust
- Public key certificates - X.509
- Certificate chains and the PKI model
- The internet PKI

Certificates and trust

Certificate concept

- A certificate is a document with the following features:
  - It is issued to some entity (think: person)
  - It asserts the entity has one or more attributes (think: date of birth, is allowed to, is working for, …)
  - It is made by some trusted issuer (think: authority)
  - It contains some means of verifying the authenticity (think: signature, watermark,…)

Certificate Example



Arab Republic of Egypt
Ministry of Communications
and Information Technology

SECC
Software Engineering
Competence Center

Certificate of Completion

This is to acknowledge that

Mohamed Mostafa Elashri

has completed

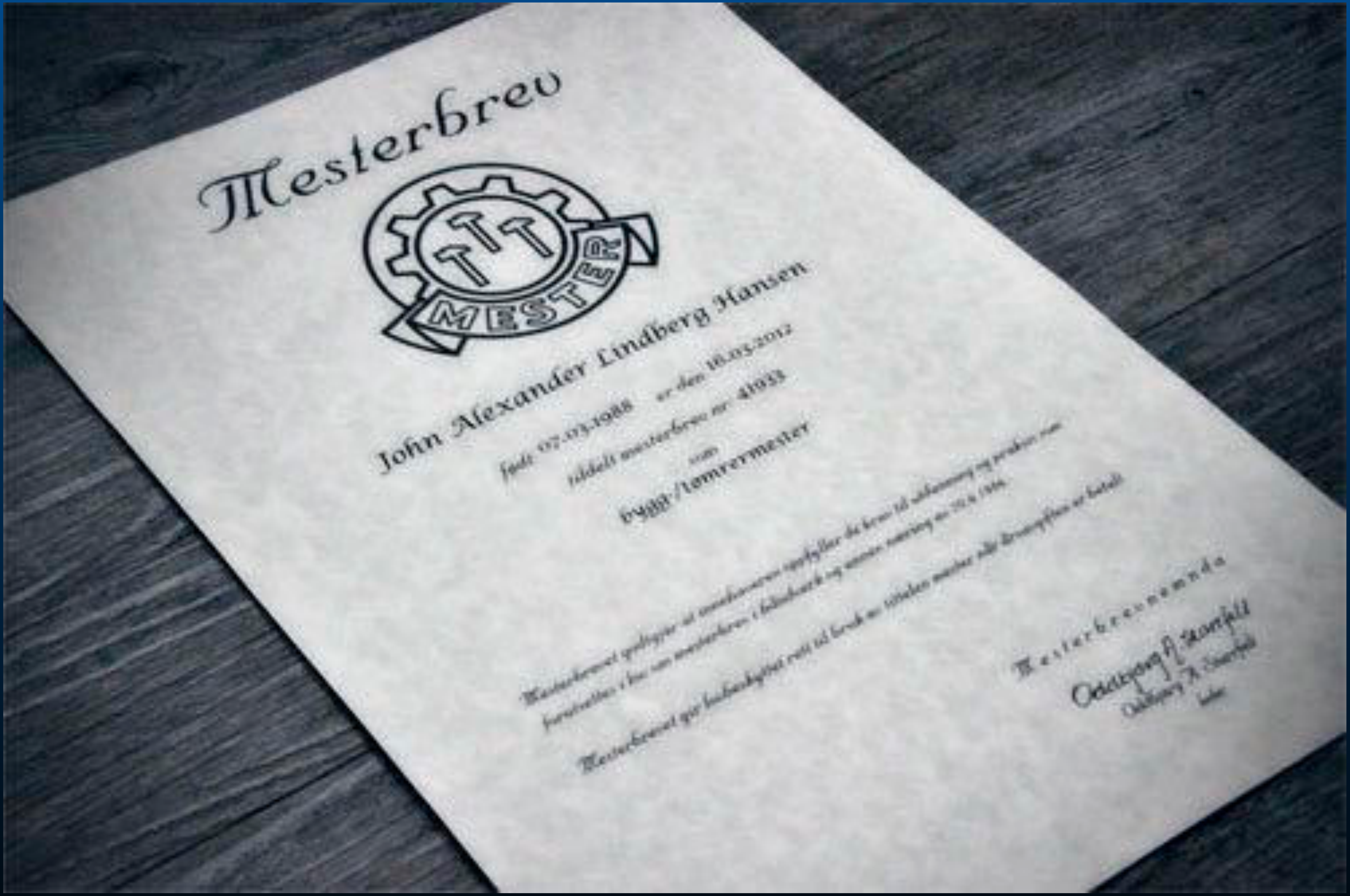
A Training Course on : **Unified Modeling Language (UML)**

held on *27 - 30 June, 2005*

Dr. Gamal M. Aly

Director

Certificate example



Mesterbrev

MESTER

John Alexander Lindberg Hansen

født 07.03.1988 er den 16.03.2012

tildelt mesterbrev nr. 41033

som

bygg/tømrermester

Mesterbrevet godtgjør at innehaveren oppfyller de krav til utdanning og praksis som forutsettes i lov om mesterbrev i håndverk og annen næring av 20.6.1986.

Mesterbrevet gir [illegible] rett til bruk av tittelen mester [illegible]

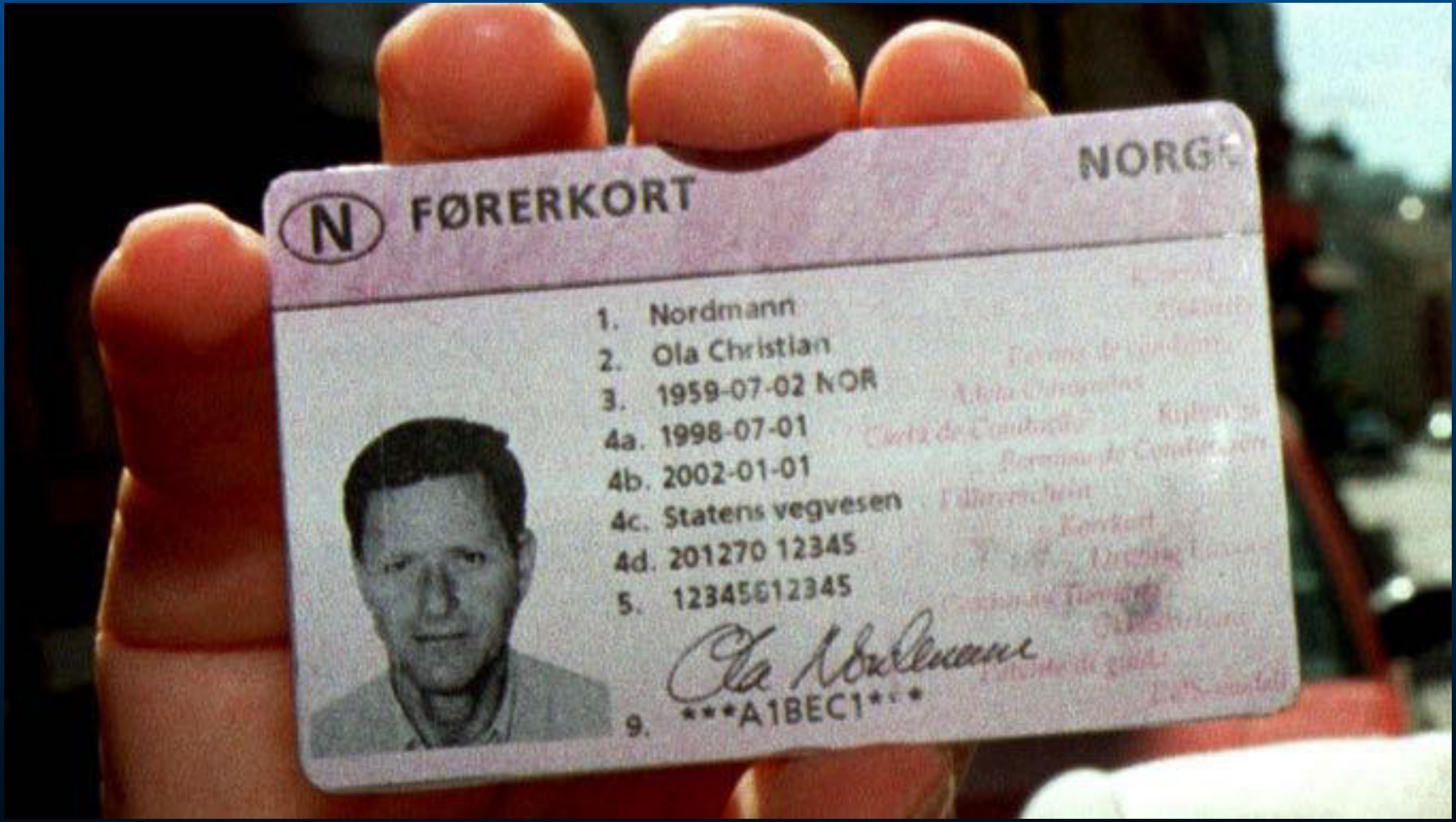
Mesterbrevnemnda

Oddbjørg A. Starrfelt

Oddbjørg A. Starrfelt

leder

Certificate example



Ⓝ FØRERKORT

NORGE

1. Nordmann
2. Ola Christian
3. 1959-07-02 NOR
4a. 1998-07-01
4b. 2002-01-01
4c. Statens vegvesen
4d. 201270 12345
5. 12345612345

Ola Nordmann

9. ***A1BEC1***

Trust

- In general, trust refers to a relationship between two parties: A (trustor) and B (trustee)

- A is said to *trust* B when A expects B to behave in some given way, and B's actions will have an influence on A

- There is always risk (for A) associated with trust if B does not behave as expected

Why do we trust?

- **Direct social relationship**. Assume friends and family will not do you any harm.
- **Earlier experiences**. Trust not betrayed earlier, assume it will not be betrayed in the future
- **Reputation**. Others claim the trustee can be trusted
- **Government**. Supposed to act in the public's best interest

Certificates and trust

- For certificates to have any meaningful use, it is necessary to trust the issuer of the certificate
- The issuer guarantees that all information on a certificate is true, in particular that the attributes applies to the entity
- Note that it is *not* necessary to trust the owner of the certificate, only the issuer and the means of verifying certificate authenticity

Public key certificates
X.509

Public/private key crypto

- Asymmetric cryptography uses two keys: one private key and one public key
- Private key should only be known to the entity owning the keys, public key can be known to anyone
- **Encryption**: Public key encrypts the message, can only be decrypted with private key
- **Digital signature**: Private key signs message, signature verified with public key

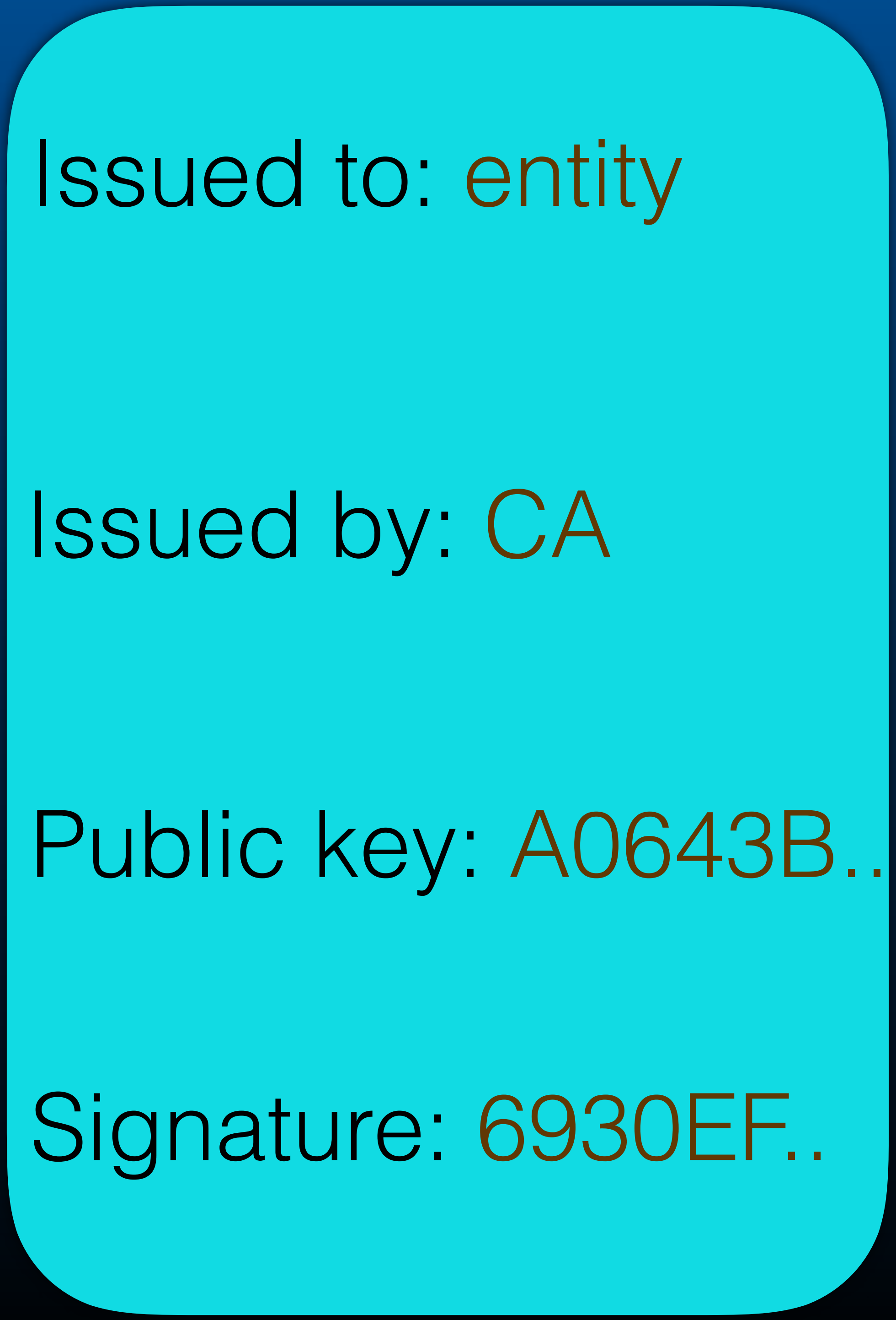
Who owns which keys?

- When encrypting a message to a particular receiver, how do you find the public key of this entity?
- Important to encrypt with the receiving entity's public key, and not an attacker's key
- Public key certificates addresses this problem: Entities get a certificate on their public key
- Certificate ties identity to public key

Public key certificates

- A public key certificate is a digital certificate:
  - The entity can be a web site, an organization, a person, etc.
  - One of the attributes is a public key
  - The issuer is a Certificate Authority (CA)
  - The certificate's authenticity is checked via a digital signature

Minimal public key certificate



Issued to: entity

Issued by: CA

Public key: A0643B..

Signature: 6930EF..

X.509

- First version of X.509 certificates defined in 1988
- Version 2 appeared in 1993, addressing some problems with reuse of names
- X.509 version 3 came in 1996, introducing flexible extensions
- The vast majority of https internet traffic is currently using X.509 v3 certificates for authenticating public keys
- X.509 contains 8 mandatory fields in addition to arbitrary many optional extension fields

Version

- Since version 1 or 2 certificates might still be around, there is a field indicating which version this certificate has

- This field should always have the value 3

Serial number

- All certificates issued by the same CA must have a unique serial number identifying a particular certificate
- CAs may adopt different policies in assigning serial numbers
- Some encode extra information in the serial number

Signature algorithm

- This field specifies which signature algorithm has been used by the CA to sign the certificate
- The signature algorithms used consist of a hash function and a public key encryption algorithm
- The most popular hash functions used are SHA1 (being phased out) and SHA256
- The most popular encryption algorithm is RSA, but elliptic curve encryption is also used

Issuer

- Unambiguous name of the CA who has issued the certificate
- The CA issuing the certificate is the entity you need to trust

Validity period

- Time period for which the certificate is valid, indicated by not-before and not-after dates

- If current date is outside of validity period, the certificate should not be used or trusted

- Validity periods typically ranges from a few months to 30+ years, depending of the type and usage of certificate

Subject name

- Unambiguous identifier of the owner of the certificate

- Subject name field is flexible, it may be

  - name, address and organisation number of a company

  - URL of a web site

  - Name of person

Subject public key info

- The public key attached to the certificate
- The CA asserts that whoever is named as subject has the private key corresponding to this public key
- In addition to value of public key, the encryption algorithm where the public key should be used is also specified

Signature value

- Generated by issuing CA, used as mean of verifying authenticity of certificate

- The complete content of the certificate, all mandatory and optional fields except for the signature value itself, is included when generating the signature

- The issuing CA’s public key is needed for verification

Extensions

- Extensions are optional, but extension fields are almost always present on X.509 certificates.
- Some extension fields have proved to be very useful in general, and are found on practically all certificates
- Extensions can be critical or non-critical:
  - If extension is critical - must be understood and processed to use certificate
  - If extension is non-critical - must be processed if understood, but OK to ignore extension field if its meaning is unknown

AuthorityInfoAccess

- Contains pointer (URL) to more information about the issuing CA
- Extension is non-critical

BasicConstraints

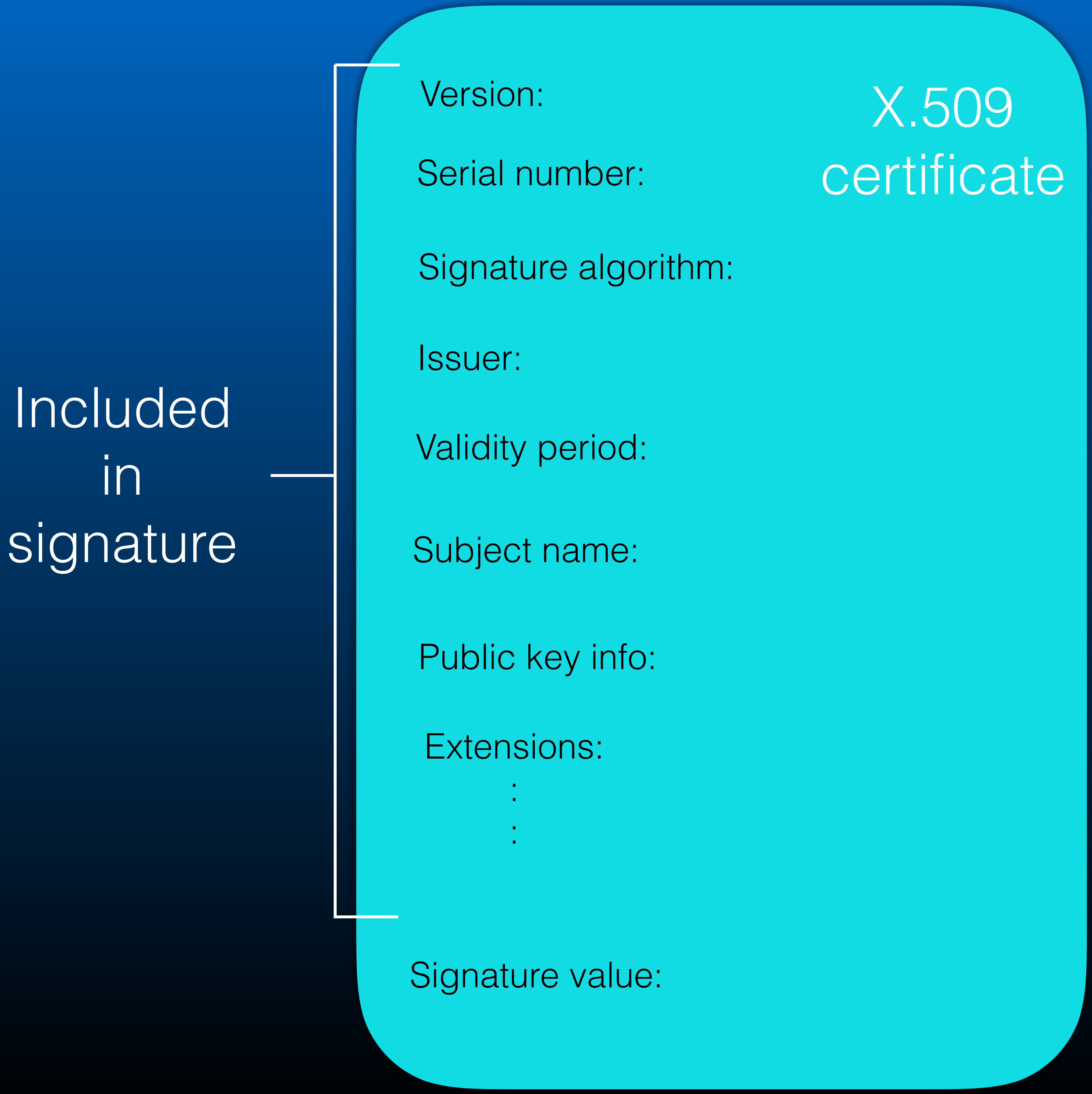
- Indicates whether this is a certificate that is used to verify signatures on other certificates
- Only CA certificates should sign other certificates
- Extension is critical in CA certificates and may or may not be critical in end-user certificates

CRLDistributionPoints

- If a certificate should be made invalid during its lifetime, it must be put on a certificate revocation list (CRL)
- Extension points to the CRL that will include this certificate, if it ever gets revoked
- Extension is normally non-critical (but CA decides)

SubjectAltName

- Field contains aliases of the owner of the certificate
- Often contains URL’s that are part of the same web site.
- Ex: certificate issued to *.wikipedia.org has wikipedia.org, *.m.wikipedia.org, *.wikimedia.org, and several others in SubjectAltName extension
- CA decides if critical or non-critical



X.509 certificate

Version:

Serial number:

Signature algorithm:

Issuer:

Validity period:

Subject name:

Public key info:

Extensions:

:

:

Signature value:

Included in signature

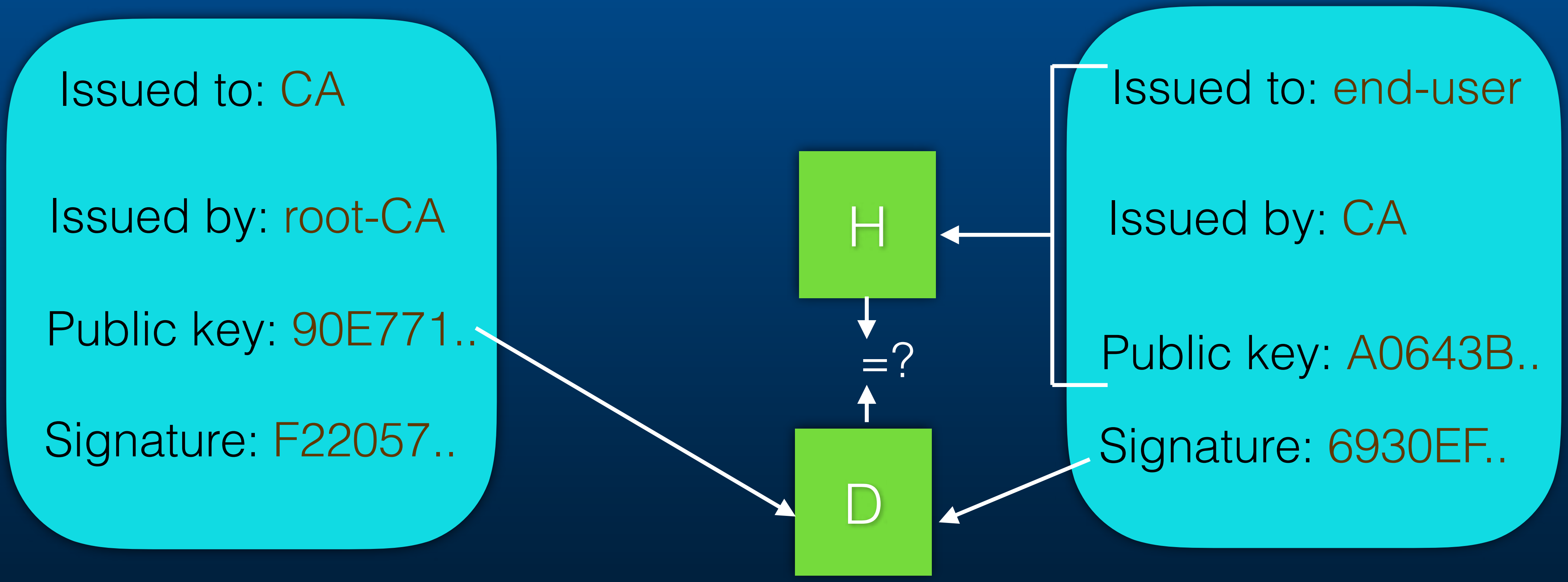
Certificate chains, trust and the PKI model

Verifying certificates

- A certificate is accepted as genuine (i.e. issued by the stated CA) if digital signature is OK

- Verifying the signature requires finding the CA’s public key

- This key is again stated on a certificate, issued to the CA



Issued to: CA

Issued by: root-CA

Public key: 90E771..

Signature: F22057..

H

=?

D

Issued to: end-user

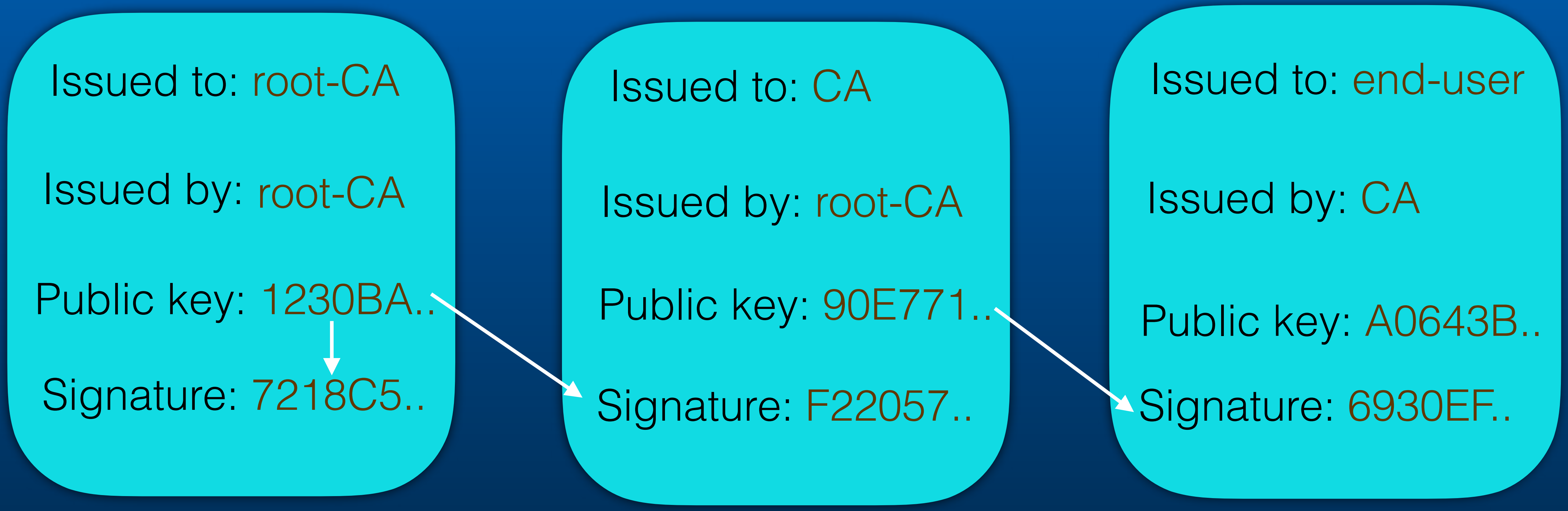
Issued by: CA

Public key: A0643B..

Signature: 6930EF..

Is CA’s certificate genuine?

- CA’s certificate is signed by root-CA
- The authenticity of CA’s certificate can be verified with root-CA’s public key
- Root-CA’s public key is, again, put on a certificate issued to root-CA



Issued to: root-CA

Issued by: root-CA

Public key: 1230BA..

Signature: 7218C5..

Issued to: CA

Issued by: root-CA

Public key: 90E771..

Signature: F22057..

Issued to: end-user

Issued by: CA

Public key: A0643B..

Signature: 6930EF..

Certificate chain: Sequence of certificates where public key on one certificate verifies the signature of the next, and the owner of one certificate is the issuer of the next

Root certificate

- A certificate chain has to start somewhere (root certificate)
- Authenticity of root certificate has to be established out of band
- Somehow we just have to «know» that the public key on the root certificate really belongs to the entity named as the root-CA
- A trusted CA is referred to as a *trust anchor* for the person trusting it

Trust root-CA(?)

- PKI model assumes that root-CAs are trust anchors by the members of the PKI

- How trust is built and established is outside the scope of the PKI model

- We must trust the root-CA to:

  - 1) Only issue certificates with true information

  - 2) Verify that the CAs it issues certificates to also only issue certificates with true information

Transfer of trust in PKI model

- Assume that you trust the root-CA, but have no relationship to the other CA

- PKI model assumes trust is transferred from the root-CA to the next CA and to the authenticity of the end-user certificate

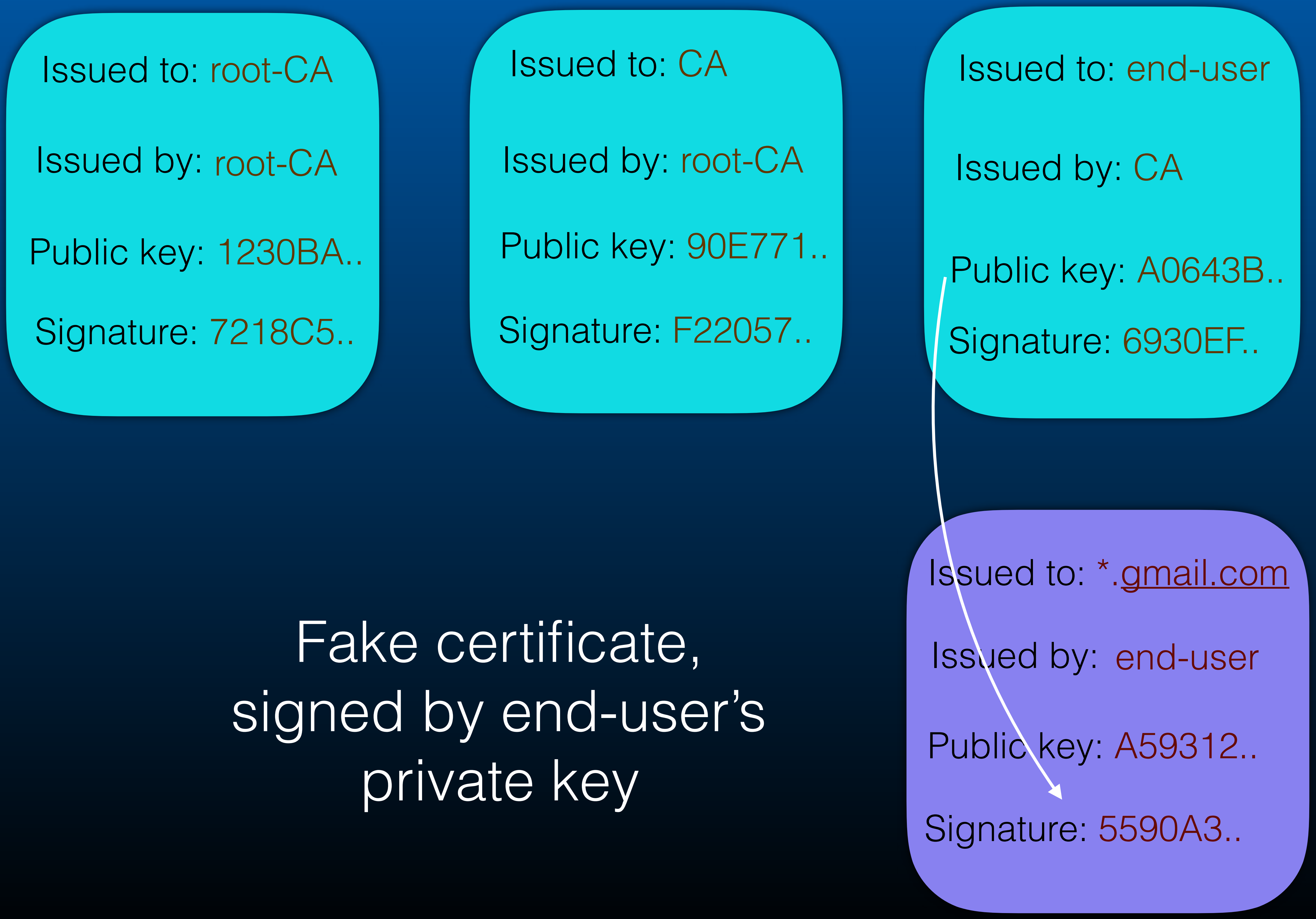
- Root-CA vouches for CA, who vouches for the end-user

- If you trust the root-CA, you can trust that the end-user has the private key corresponding to the public key on the end-user certificate

Early attack on certificate chain

- Why is the BasicConstraints extension needed?
- Recall, BasicConstraints indicates whether a certificate can be used to verify signatures on other certificates
- Earlier PKI implementations did not consider this, leading to a successful man-in-the-middle attack

Genuine certificate chain, endorsed by root-CA



Issued to: root-CA

Issued by: root-CA

Public key: 1230BA..

Signature: 7218C5..

Issued to: CA

Issued by: root-CA

Public key: 90E771..

Signature: F22057..

Issued to: end-user

Issued by: CA

Public key: A0643B..

Signature: 6930EF..

Issued to: *.gmail.com

Issued by: end-user

Public key: A59312..

Signature: 5590A3..

Fake certificate,
signed by end-user's
private key

Attack

- Attacker acts as man-in-the-middle, masquerading as www.gmail.com

- Client connects to attacker, and is served the chain of four certificates

- If BasicConstraint is not set or processed, client will accept attacker as www.gmail.com, based on trusting the root-CA

The Internet PKI

Secure internet communication

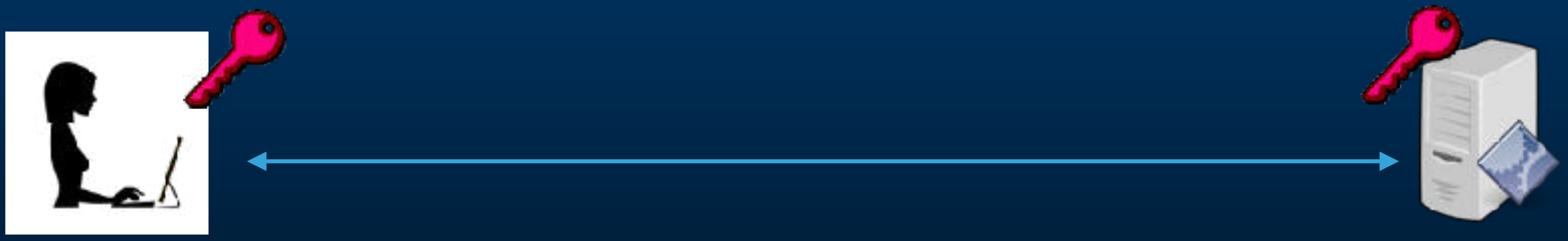
- TLS is the dominating standard for secure web traffic on the internet
- TLS is based on PKI and supports mutual authentication of client and server
- Most often, only servers are authenticated using PKI, while clients are authenticated in other ways

Communication secured by TLS

- Two entities want to communicate «securely» over the internet
- More specifically, they want
  - Encryption - no one else can learn the content of messages
  - Integrity - information remains unchanged in transit
  - Authentication - the two parties know who they are communicating with

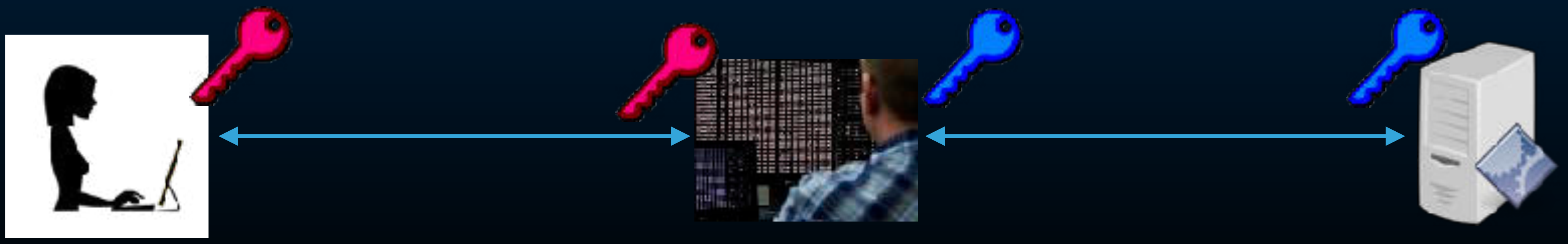
Easy and hard problems

- Easy: encryption and integrity (thanks to DH key exchange)
- Hard: authentication (reason for whole PKI machinery)
  - How to separate



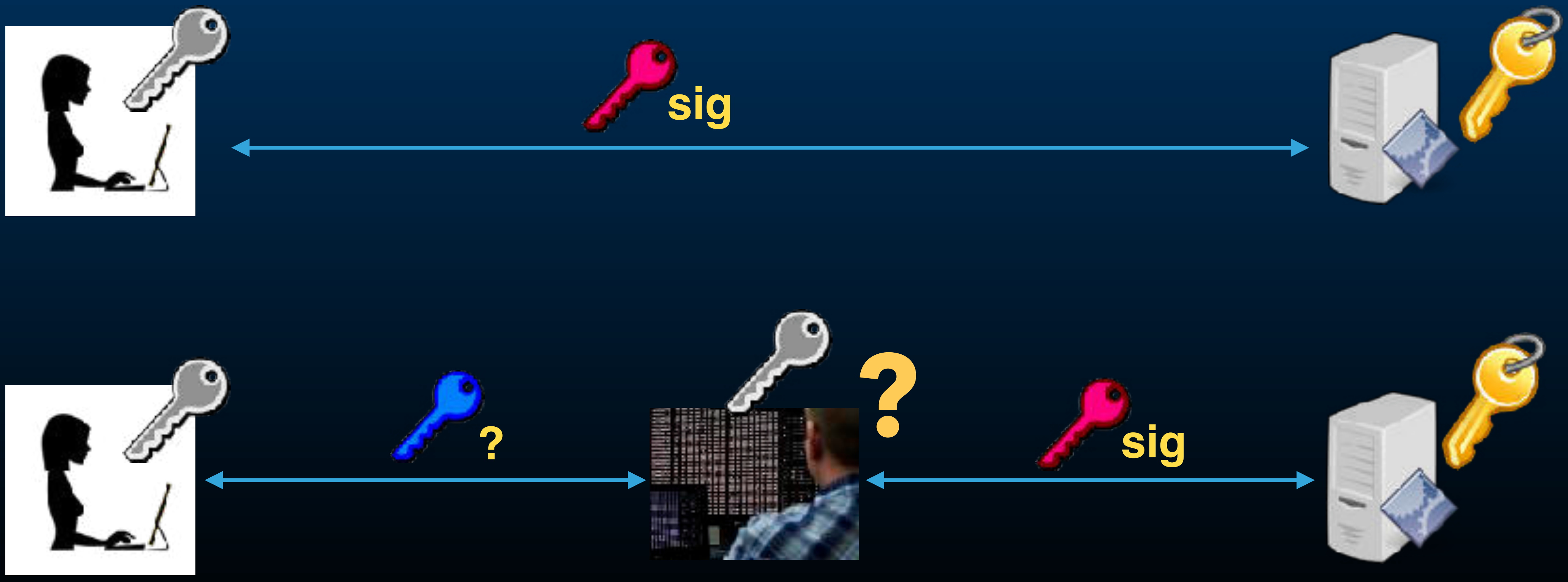
from

?



Authentication by PKI

- Server has private key, client knows public key
- Server signs some data (f.ex. the DH exchanged secret key) with its private key



Web site authentication

- When connecting to a web server over TLS, authentication happens automatically, with no user interaction

- Web browsers come with many root certificates pre-installed

- Trust in the root-CAs is implicitly made by the browser, for the user

Web site authentication

- Browser receiving a certificate chain checks that…
  - chain starts with a pre-installed root certificate
  - the signatures are OK throughout the chain
  - each certificate is within its validity period
  - no certificates have been revoked
  - all critical extensions are processed
- If all checks are OK, web site is authenticated

Trust?

- Trust is an important ingredient in the PKI model
- PKI model assumes each user makes a choice of which CAs to trust
- In the internet PKI, web browsers has made a choice of which CAs to trust, and users are trusting the browser (whether they know it or not)

Model vs reality

- The ‘trust’ part of the PKI model does not match the reality on the internet today
- The vast majority of users do not know the root-CAs acting as their trust anchors, or what a CA is
- The PKI model is good, but assumes that each user understands the model and makes conscious choices about which CAs to trust