


Public Key Infrastructure Fundamentals

Prof. Bart Preneel
 COSIC – KU Leuven - Belgium
 Firstname.Lastname(at)esat.kuleuven.be
 http://homes.esat.kuleuven.be/~preneel
 March 2016

Thanks to Paul van Oorschot


1



Goals

- Understand how public keys can be distributed and revoked on a large scale
- Understand what a CA-based PKI is and what the problems are with their deployment
- Understand how multiple CAs can interoperate depending on their trust relationship


2



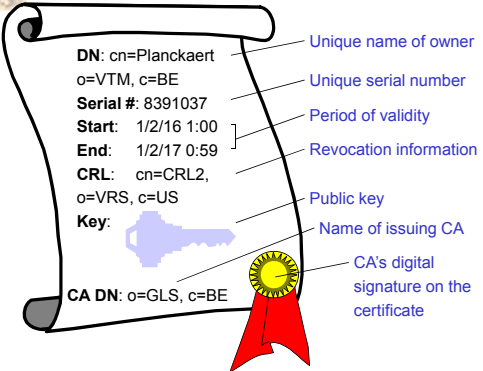
How to establish public keys?


- point-to-point on a trusted channel
 - *mail business card, phone*
- direct access to a trusted public file (registry or database)
 - *authentication trees*
- on-line trusted server (bottleneck)
 - *OCSP: Online Certificate Status Protocol*
- off-line servers and certificates
 - *PKI: Public Key Infrastructure*
- implicit guarantee of public parameters
 - *identity based and self-certified keys*

3




What is a Certificate?



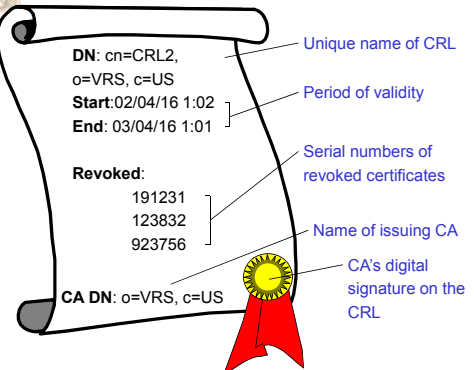
DN: cn=Planckaert
 o=VTM, c=BE
Serial #: 8391037
Start: 1/2/16 1:00
End: 1/2/17 0:59
CRL: cn=CRL2,
 o=VRS, c=US
Key: 
CA DN: o=GLS, c=BE

- Unique name of owner
- Unique serial number
- Period of validity
- Revocation information
- Public key
- Name of issuing CA
- CA's digital signature on the certificate

4




What is a Certificate Revocation List?



DN: cn=CRL2,
 o=VRS, c=US
Start: 02/04/16 1:02
End: 03/04/16 1:01
Revoked:
 191231
 123832
 923756
CA DN: o=VRS, c=US

- Unique name of CRL
- Period of validity
- Serial numbers of revoked certificates
- Name of issuing CA
- CA's digital signature on the CRL


5



PKI Overview

1. Background: Keys and Lifecycle Management
2. PKI components ("puzzle pieces")
3. Trust Models

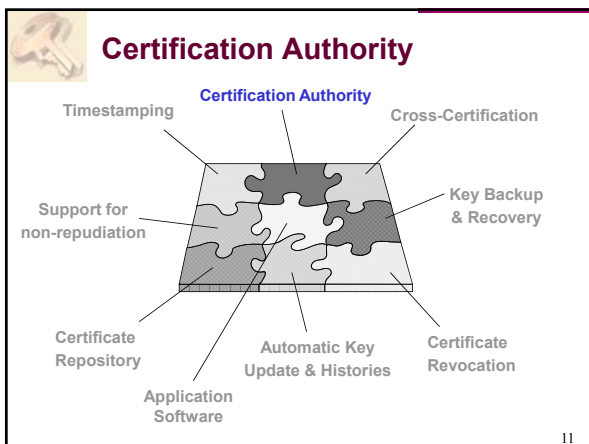
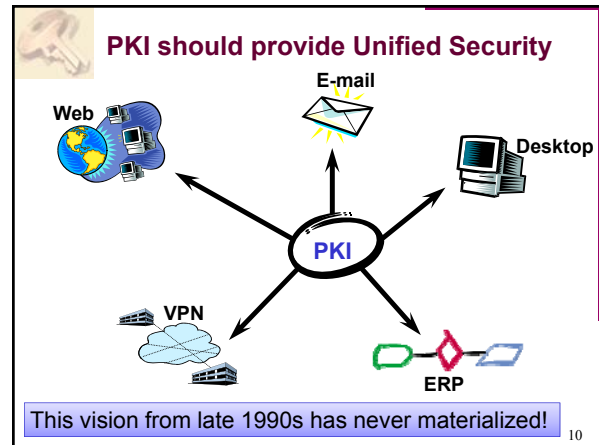
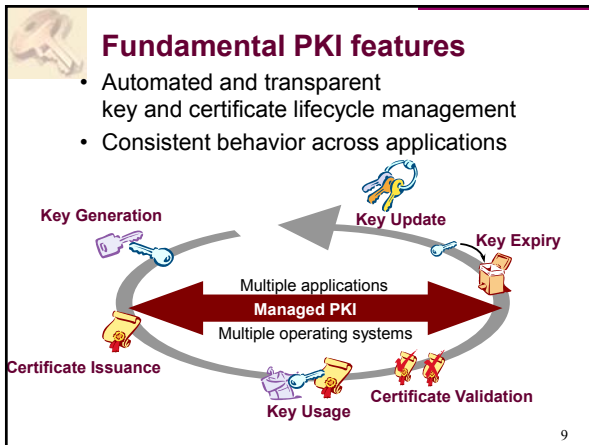
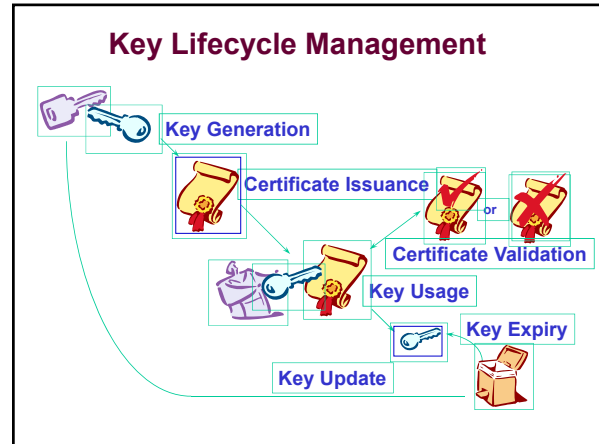
6



Background:

Keys and Lifecycle Management

7



- ### Certification Authority
- Issue certificates for all entities / devices (for multiple applications) from a single CA
 - single system saves h/w, s/w, training, personnel
 - Flexible certificate policy / security policy
 - tailor to needs of environment, application or entity (e.g. certificate lifetime, crypto algorithms, keylengths, password rules, ...)
- 12

Certificate Repository

A diagram showing a trapezoidal shape composed of several interlocking puzzle pieces. The pieces are labeled with various PKI components: Timestamping, Certification Authority, Cross-Certification, Key Backup & Recovery, Certificate Revocation, Automatic Key Update & Histories, Application Software, Certificate Repository, and Support for non-repudiation. The 'Certificate Repository' piece is highlighted in blue.

13

Certificate Repository

- LDAP-compliant directory stores certificates
 - standards-based for interoperability
- Directory products built specifically to address scalability issues
 - X.500 or proprietary schemes to replicate data (scales to millions of users)

14

Certificate Revocation System

A diagram showing a trapezoidal shape composed of several interlocking puzzle pieces. The pieces are labeled with various PKI components: Timestamping, Certification Authority, Cross-Certification, Key Backup & Recovery, Certificate Revocation, Automatic Key Update & Histories, Application Software, Certificate Repository, and Support for non-repudiation. The 'Certificate Revocation' piece is highlighted in blue.

15

Certificate Revocation

- Automated CRL publishing
 - when certificate revoked, CRL can be automatically published to directory providing near-immediate availability
 - automated CRL checking by application
 - want to avoid applications which require manual end-user actions to check CRLs for each application or certificate usage

March 2001: Verisign has issued two certificates to fake Microsoft employees

- Problem: IE did not implement revocation checking

16

Automated Key Update & History

A diagram showing a trapezoidal shape composed of several interlocking puzzle pieces. The pieces are labeled with various PKI components: Timestamping, Certification Authority, Cross-Certification, Key Backup & Recovery, Certificate Revocation, Automatic Key Update & Histories, Application Software, Certificate Repository, and Support for non-repudiation. The 'Automated Key Update & Histories' piece is highlighted in blue.

17

Automated Key Update & History

- Users should never even need to know they have their own certificates (password only)
- If key management is not automated or does not provide key history . . .
 - when certificate expires, lose access to all past encrypted data, e-mail, . . .
 - user must request new certificate and repeat entire registration process
- Should replace key, not just new expiry date
- Transparent triggering mechanism

18

Key Backup & Recovery

19

Key Backup & Recovery

- Enterprise will lose valuable (stored) data if keys used to encrypt data are not backed up
 - 20-40% of users forget passwords / year
 - employees leave the organization
- Allows the enterprise to control the backup
 - not reliant on 3rd parties
 - should be configurable to require multiple administrators to authorize access

Key recovery/backup for **storage** keys should not be confused with **key escrow**; governments have tried to impose this for encryption keys used **for communication**

20

Support for Non-Repudiation

21

Support for Non-Repudiation

- Must use separate key pairs for digital signatures and encryption
 - want backup of encryption keys, **do not** want backup of signature private keys
- Separate key pairs allows lifecycles to be managed independently
- Different policy controls for each key pair
 - security requirements per pair may differ, e.g. valid lifetimes

22

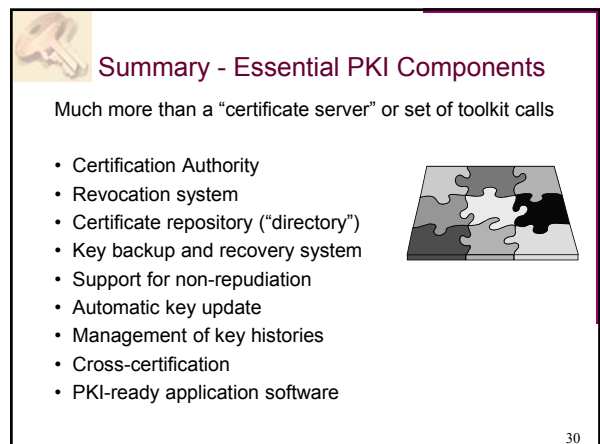
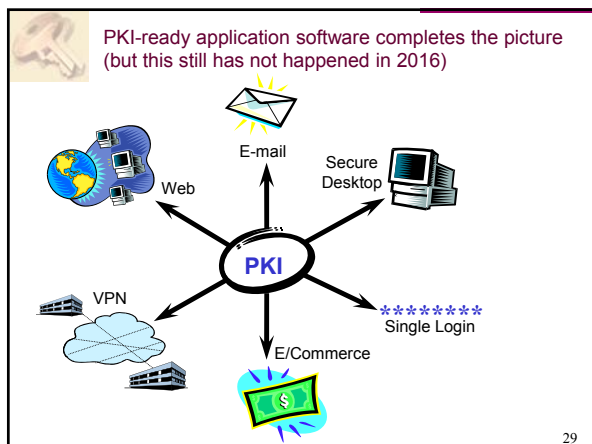
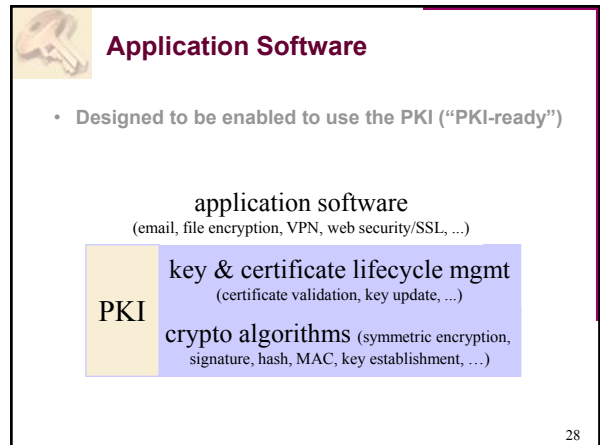
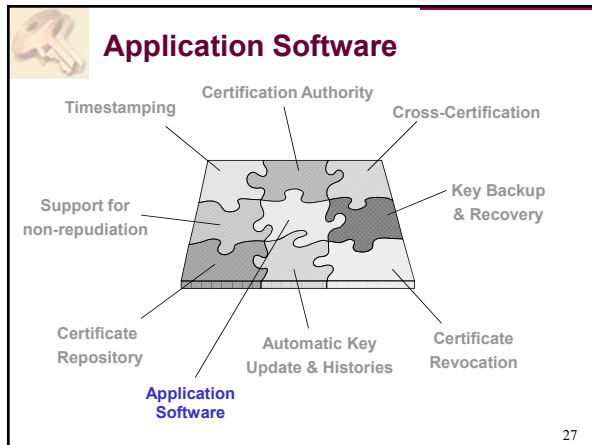
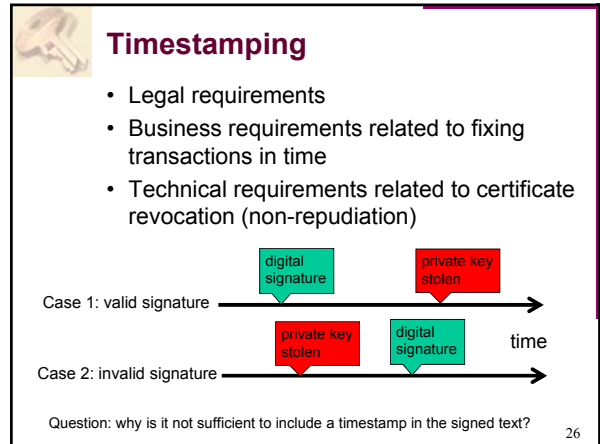
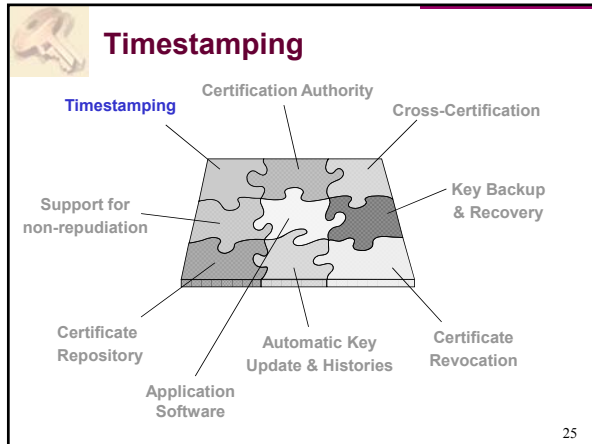
Cross-Certification

23

Cross-Certification (cf. Trust models)

- Sufficiently flexible to model existing business relationships
 - includes 1-1 relationships and hierarchies
 - cross-certificate associated with an organization (vs. a service provider)
 - compare to web trust model: trust anyone signed by browser-embedded CAs
- Enterprise manages cross-certification policy & procedures, to reduce business risk
 - cross-certificates created by authorized administrators, transparent to end-user

24






More info: IETF PKIX Working Group

www.ietf.org

- de facto standards for Internet PKI, X.509-based
- Certificate & CRL Profile [PKIX-1]: RFC 2459
- Certificate Mgmt Protocols [PKIX-CMP, PKIX-3]: RFC 2510
- PKIX roadmap: www.ietf.org/internet-drafts/draft-ietf-pkix-roadmap-01.txt


31



PKI vs. Privilege Management

- Public key certificate binds a public key to an entity
- Establishes who owns a key vs. what privileges that key / owner is granted
- Certificate-processing software (relying party) may implicitly grant privileges
- Privilege Management Infrastructure (PMI) makes privileges explicit
- PMI may utilize PKI as base infrastructure
- example: attribute certificates


32



Key generation: where?


- CA generates key for user
 - absolute trust
 - need transport of private keys
 - easier management for backup/recovery
- user generates his/her key
 - does user have the expertise? (ok if smart card)
 - need to transport of public keys (integrity channel)
- specialised third party generates keys

33



Trust Models


34



Hierarchical trust model

Relying parties transfer risk to the Root CA

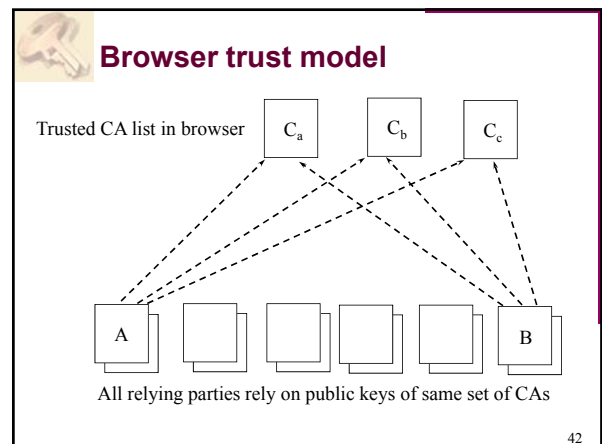
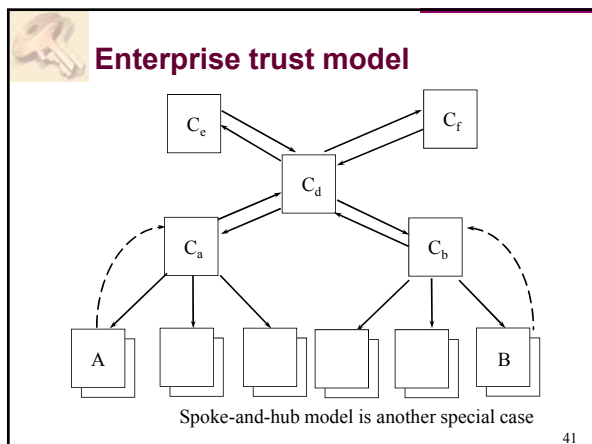
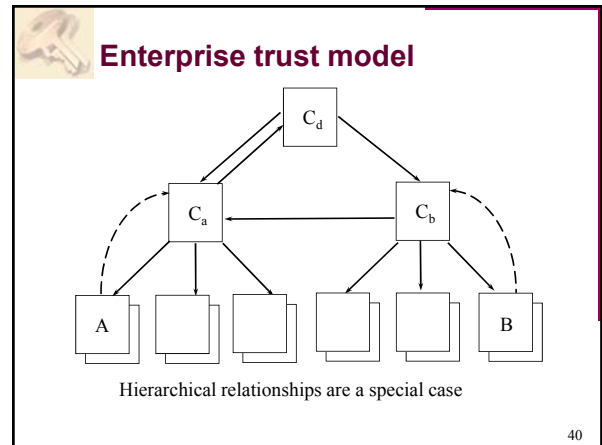
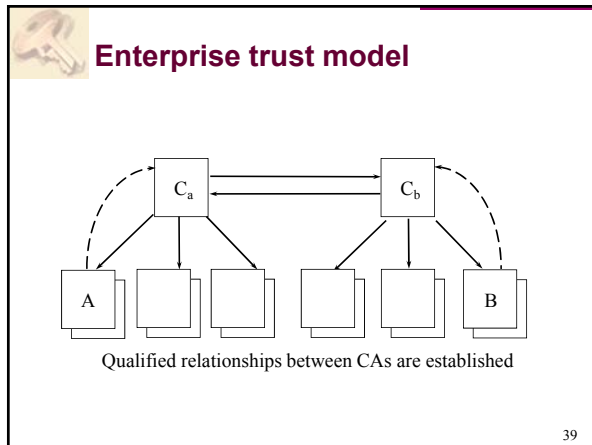
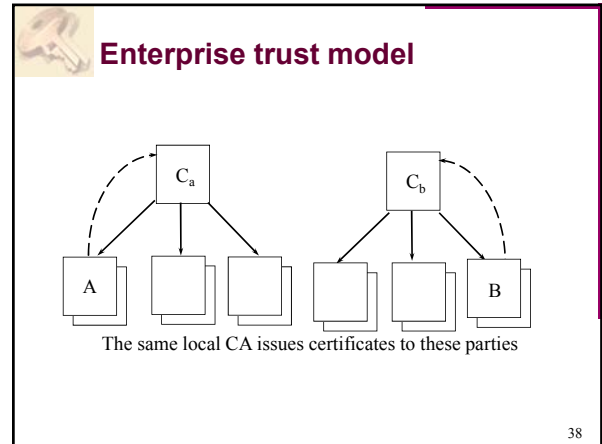
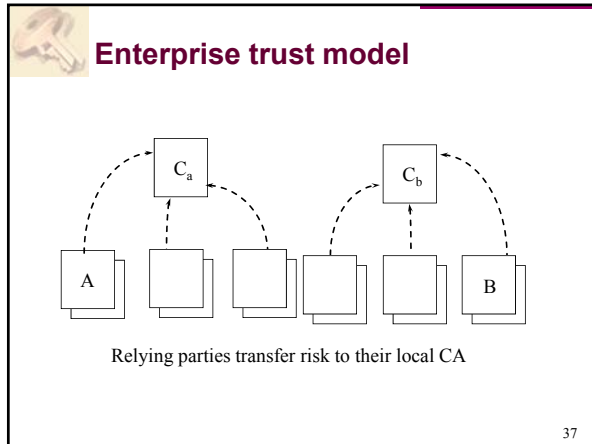
35

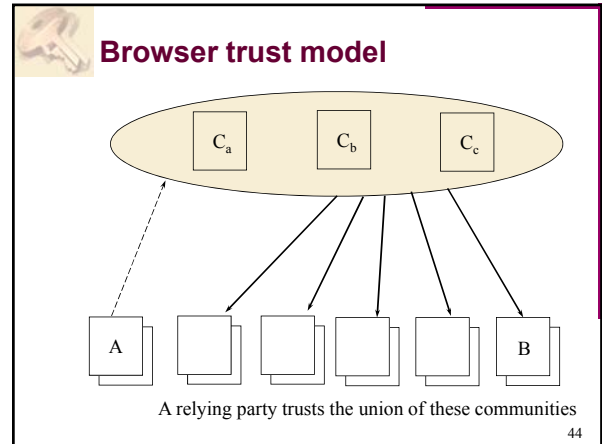
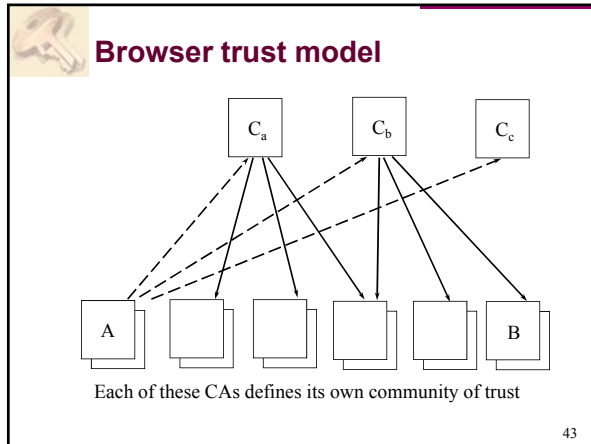


Hierarchical trust model

Root CA “deputizes” subordinate CAs, which issue certificates

36





The CA Mess on the web

[Eckersley10] "An observatory for the SSLiverse"

- 10.8M servers start SSL handshake
- 4.3M use valid certificate chains
- 650 CA certs trustable by Windows or Firefox (industry: only 65 main)
- 1.4M unique valid leaf certs
 - 300K signed by one GoDaddy cert
- 80 distinct keys used in multiple CA certs
- several CAs sign the IP address 192.168.1.2 (reserved by RFC 1918)
- 2 leaf certs have 508-bit keys
- Debian OpenSSL bug (2006-2008)
 - resulted in 28K vulnerable certs
 - fortunately only 530 validate
 - only 73 revoked

How can we fix this mess?

45

CA incidents

- March 2011 – Comodo: 9 fraudulent certs
 - via RA GlobalTrust.it/InstantSSL.it
- Summer 2011 – DigiNotar: 500+ fraudulent certs
 - meet-in-the-middle attack against Google users in Iran (300K unique IPs, 99% from Iran)
 - filed for bankruptcy 20 September 2011
- January 2013 – Turktrust CA incident
- February 2013 – Bit9 lost signing key
- Recent incidents: CCA (India), CCNC and Lenovo (China), ANSSI (France), Symantec
- Products adding trusted roots in trust store
 - Lenovo incident
 - Interception of social media usage by employers

46

CA incidents

- Malware signed by key of Government of Malaysia

47

Mobile CA

- O/S trust store
 - many Android phones run old versions and have old Trust Store
 - Android Pre-2.3 does not support SHA-256
 - still certs with MD5 and SHA-1
- Mobile Apps
 - ALLOW_ALL_HOSTNAME: 35% of apps; e.g., Facebook, Baidu
 - Custom Trust Store: not always better

<https://bluebox.com/technical/trust-managers>

48

Improvements to CA ecosystem

- DANE – based on DNSSEC – specify restrictions for a given SSL/TLS server
 - would need hard fail
- CA Authorization (RFC 6844): tell CA - if you are not one of the CAs on this list, don't issue certs for this domain (competition issue?)
- Pinning: tell clients - cert for this site look like this; if you detect something else, this may be a breach (more likely a misconfiguration)
 - not for "small" sites? (need bootstrap)
 - seems to work for Google/Chrome ecosystem
- Cert Transparency: certs public in authenticated tree

49

CA common problem

50

Personal trust model (and related: "web-of-trust")

- all entities are end-users (CAs do not exist)
- keys are essentially self-guaranteed
- some end-users may also be *introducers*
- end-user imports public keys of others

CHARACTERISTICS

- suits individuals, not enterprise/corporations
- user-centric
- requires security-aware end-users
- poor scalability

51

PGP/GPG Key Servers

- Centralized support for web of trust: servers that hold huge public key rings
 - update to each other, accept and send updates from/to everyone
 - better than everyone keeping a huge key ring
 - server addresses included with PGP/GPG software
 - concerns: privacy, user registration/verification (are you Bill Gates?) and key revocation

Example: PGP Global Directory

52

Trust models & Revocation

- public-key systems are commonly engineered with long-life certificates
- certificates bind a key-pair to identity (and potentially privilege information)
- circumstances change over certificate life
 - keys may become compromised
 - identifying information may change
 - privilege may be withdrawn
- need ability to terminate the binding expressed in the certificate
- revocation: most difficult issue in practice

53

Revocation options

mechanisms indicating valid certificates

- short-lifetime certificates


mechanisms indicating invalid certificates

- certificate revocation lists - CRLs (v1 X.509)
- CRL fragments (v2 X.509), including ...
 - segmented CRLs (CRL distribution points)
 - delta CRLs
 - indirect CRLs

mechanisms providing a proof of status

- status-checking protocols (OCSP, ValiCert)
- iterated hash schemes (Micali)
- certificate revocation trees


54



CRL: properties

- basic CRL
 - simplicity
 - high communication cost from directory to user
- improved CRL
 - very flexible
 - more complex
 - reduced communication and storage


55



Online Certificate Status Protocol (OCSP) [RFC 2560]

- on-line query to
 - CA
 - or Trusted Responder
 - or CA designated responder
- containing
 - hash of public key CA
 - hash of public key in certificate
 - certificate serial number


56



OCSP: signed answer

- status
 - good: not revoked
 - revoked
 - unknown
- time
 - thisUpdate
 - nextUpdate
 - producedAt

57




OCSP: evaluation

- [+] positive and negative information
- [-] need to be on-line
 - risk for denial of service
 - not always possible
- ! OCSP may send you **freshly signed but old** information

If a browser gets **no answer** to an OCSP request, it just goes on as if nothing happened (usability is more important than security)

<http://blog.spiderlabs.com/2011/04/certificate-revocation-behavior-in-modern-browsers.html>


58



Revocation summary

- established standards for basic revocation
 - ITU-T X.509: 1997, ISO/IEC 9594-8: 1997
 - v2 CRLs
- more sophisticated solutions may be needed for specific applications
- revocation of higher level public keys is very hard (if not impossible)
 - e.g. requires browser patch
- even after 20 years of PKI history, revocation is problematic in practice

59



Characterizing questions for trust models

- what are the types/roles of entities involved
- who certifies public keys
- are trust relationships easily created, maintained, updated
- granularity of trust relationships
- ability of particular technology to support existing business models of trust
- how is revocation handled?
 - . . . of end-users . . . of certification authorities

60

Trust model continuums

hierarchical browser enterprise personal
[increasing granularity of trust]

hierarchical browser personal enterprise
[increasing capability to represent B2B trust]

Many other continuums can be formulated

61

Trust model summary

Key idea: manageability of trust relationships
Each model has its place --

- personal trust model: okay for security-aware individuals working in small communities
- browser model: simple, large communities, everyone trusts all CAs defined by s/w vendor
- hierarchical model: best given an *obvious* global root and a *grand design* methodology
- enterprise trust model: best between peer organizations, where trust flexibility is required
- global PKI will include variety of trust models

62

Identity based encryption

- Extra material for information

63

Identity-Based Encryption (IBE)

- IBE is an old idea
 - Originally proposed by Adi Shamir, S in RSA, in 1984
 - Not possible to build an IBE system based on RSA
- First practical implementation
 - Cocks IMA 2001 and Boneh-Franklin Algorithm Crypto 2001
 - Bilinear Maps (Pairings) on Elliptic Curves
 - Based on well-tested mathematical building blocks
 - Public Key Algorithm used for Key Transport
- The IBE breakthrough is having major impact
 - Now over 400 scientific publications on IBE and Pairing Based Cryptography
 - Major deployments in industry
- Standardization Efforts
 - IBE mathematics is being standardized in IEEE 1363.3
 - IETF S/MIME Informational RFC

64

IBE Public Keys

... Introduce This Elegance

Public-key Encryption where Identities are used as Public Keys

- IBE Public Key:
alice@gmail.com
- RSA Public Key:

Public exponent=0x10001
Modulus=13506641086599522334960321627880596993888147
560566702752448514385152651060485953383394028715
057190944179820728216447155137368041970396419174
304649658927425623934102086438320211037295872576
235850964311056407350150818751067659462920556368
552947521350085287941637732853390610975054433499
9811150056977236890927563

65

How IBE works in practice

Alice sends a Message to Bob

Key Server
• Master Secret
• Public Parameters

2 Requests private key, authenticates

3 Receives Private Key for bob@b.com

alice@a.com bob@b.com

1 Alice encrypts with bob@b.com

4 Bob decrypts with Private Key

66

How IBE works in practice

Alice sends a Message to Bob

Key Server

Fully off-line - no connection to server required

1 Charlie encrypts with bob@b.com

2 Bob decrypts with Private Key

67

IBE Public Key Composition

v2 ||
public key definition version

ibe-server.acme.com#1234 ||
server location and public parameter version

week = 252 ||
key validity period

bob@acme.com
e-mail address

68

IBE Benefits

Dynamic "As Needed" Public and Private Key Generation

- No pre-generation or distribution of certificates
- Built-in Key Recovery - No ADKs
- Allows content, SPAM, and virus scanning at enterprise boundary
- Facilitates archiving in the clear per SEC regulations

Policy in the Public Key

- e.g. Key Validity Period
- No CRLs

Dynamic Groups

- Identities can be groups and roles; no re-issuing keys when group or role changes

Minimal System State

- Master Secret / Public Parameters (~50KB) all you need for disaster recovery
- End user keys and message not stored on server
- Server scalability not limited by number of messages

Benefits claimed to lead to:

- High system usability
- Highly scalable architecture
- Low operational impact
- Fully stateless operation

69

Public Key Infrastructure

Certificate Server binds Identity to Public Key

68

Identity Based Encryption

Binding of Identity to Key is implicit

69

IBE summary

- Sounds cool
- Lack of revocation means short-lived keys hence high overhead for recipient
- Key escrow is problematic (definitely for signatures)
 - can be avoided but only with a complex scheme that needs PKI anyway
- How do you know what the system parameters used by people with the address xx@hotmail.com?
 - Can these system parameters be revoked?

72



PKI

- Public key cryptography and public keys are essential for large scale secure systems
- PKI as we know today is designed for an off-line world in 1978
- Global PKI is very hard
 - who is authoritative for a given namespace?
 - liability challenge
- Revocation is always hard

- Things are much easier if relying party is the same as issuing party: no certificates are needed

73