IT-Security (ITS) B1

DIKU, E2025

Today’s agenda

Recap

Key exchange

Key management

Certificates

Recap: Security goals and crypto primitives

Don’t worry about the details of RSA, AES, or SHA1

Focus on the bigger picture of what we achieve with

- symmetric and asymmetric ciphers
- cryptographic hash functions
- message authentication codes
- digital signatures

Key management

Many keys to protect

Master key

Session key

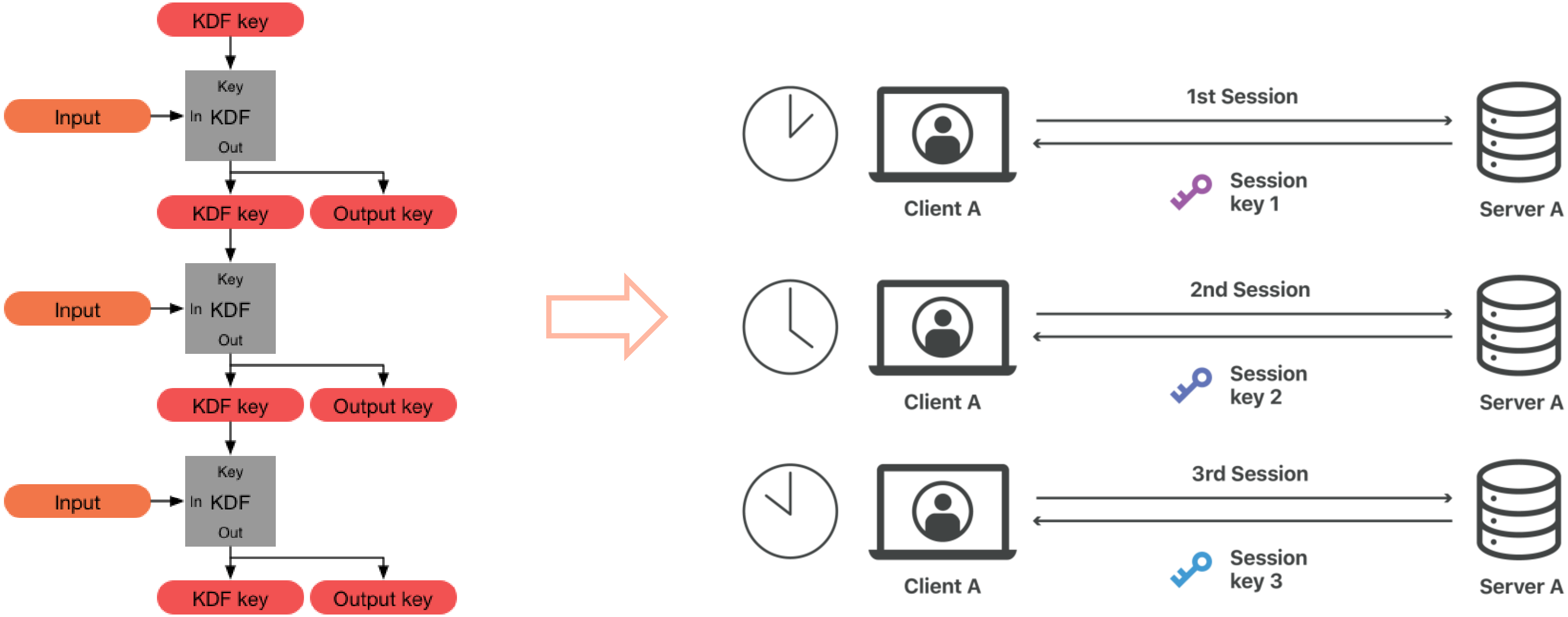
Signature key

Data encryption key

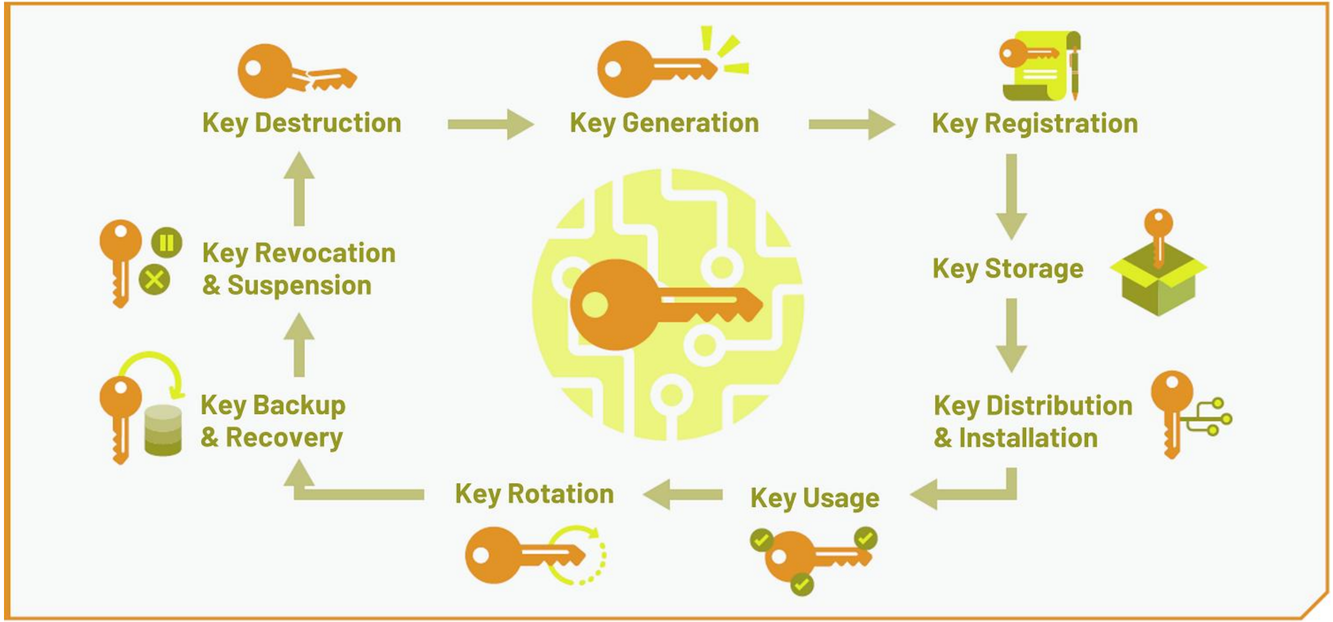
Key encryption key

...

Key derivation functions and sessions keys



Protect during entire lifecycle



Key exchange

Key exchange options include

Pre-distribution

Generated and distributed “ahead of time” e.g. physically

Distribution

Generated by a trusted third party (TTP) and sent to all parties

Agreement

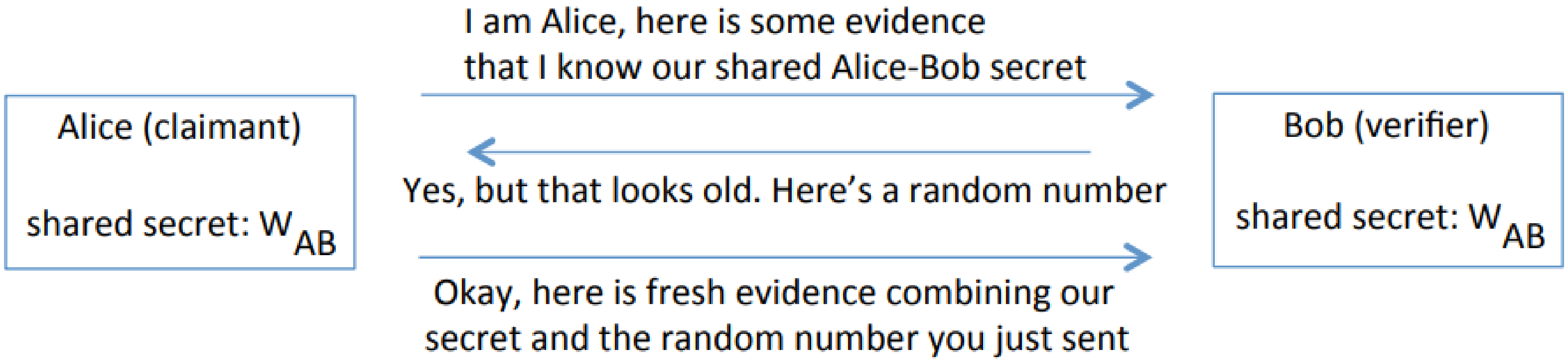
Generated by all parties working together

Asymmetric

Is e really yours?

Key distribution

Basic authenticated key exchange

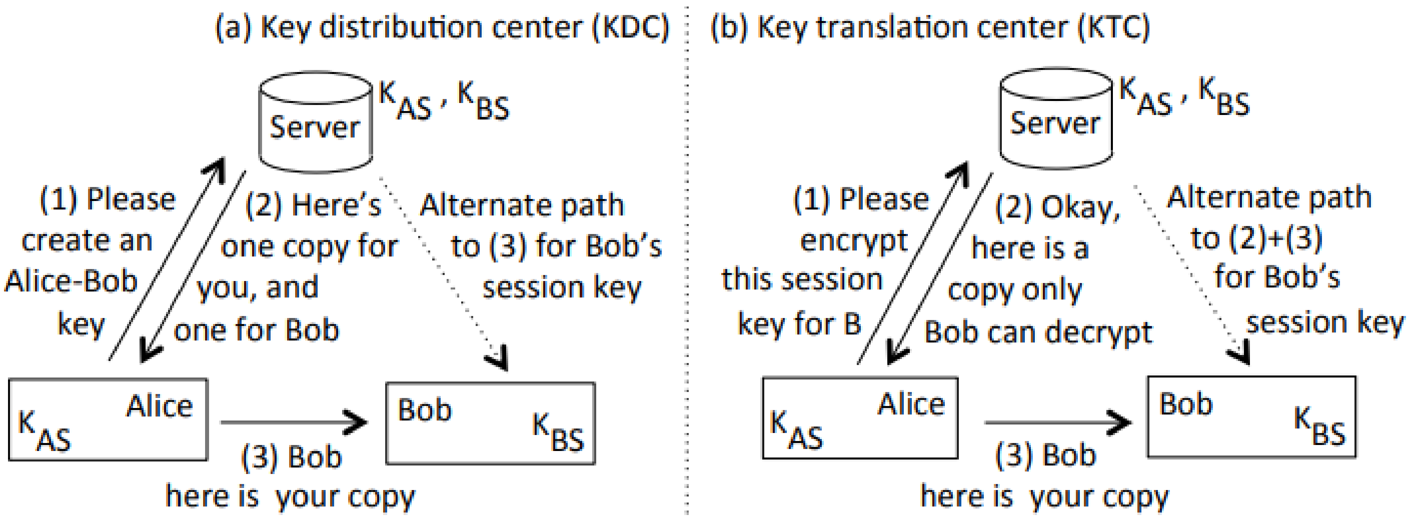


I am Alice, here is some evidence that I know our shared Alice-Bob secret

Yes, but that looks old. Here’s a random number

Okay, here is fresh evidence combining our secret and the random number you just sent

With a trusted third party


(a) Key distribution center (KDC)
Server
K_{AS}, K_{BS}
(1) Please create an Alice-Bob key
(2) Here's one copy for you, and one for Bob
Alternate path to (3) for Bob's session key
K_{AS}
Alice
Bob
K_{BS}
(3) Bob here is your copy
(b) Key translation center (KTC)
Server
K_{AS}, K_{BS}
(1) Please encrypt this session key for B
(2) Okay, here is a copy only Bob can decrypt
Alternate path to (2)+(3) for Bob's session key
K_{AS}
Alice
Bob
K_{BS}
(3) Bob here is your copy

Developing a key distribution scheme

Situation:

A and B want to exchange keys remotely

Both A and B share a key (K_AS, K_BS) with a trusted third party, S

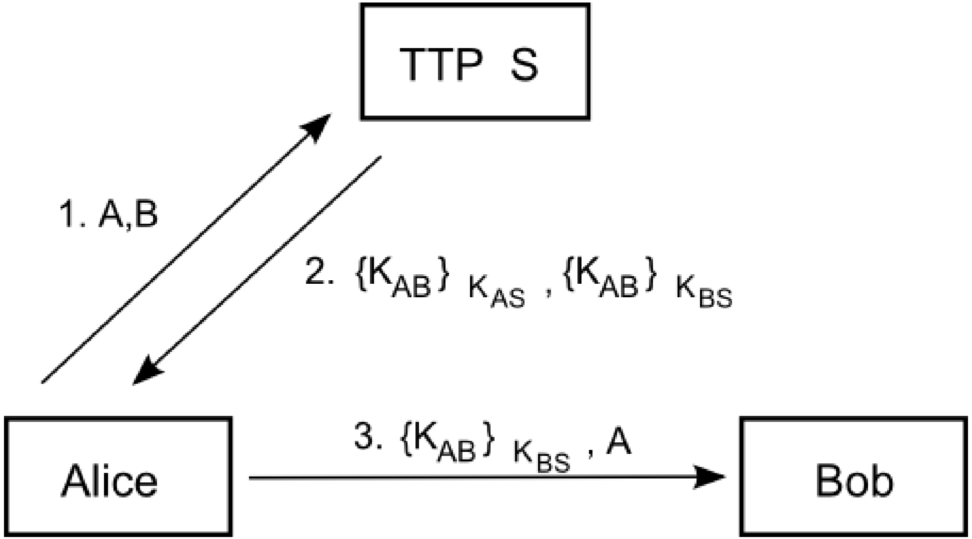
At the end, we want to achieve:

A and B know a new key K_AB

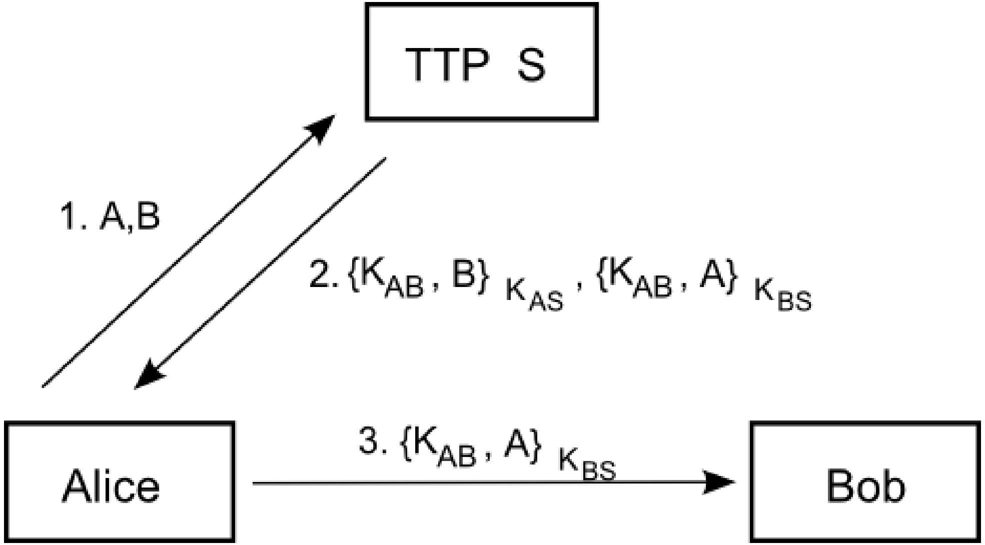
No one but A, B, and possibly S knows K_AB

A and B know that K_AB is newly generated

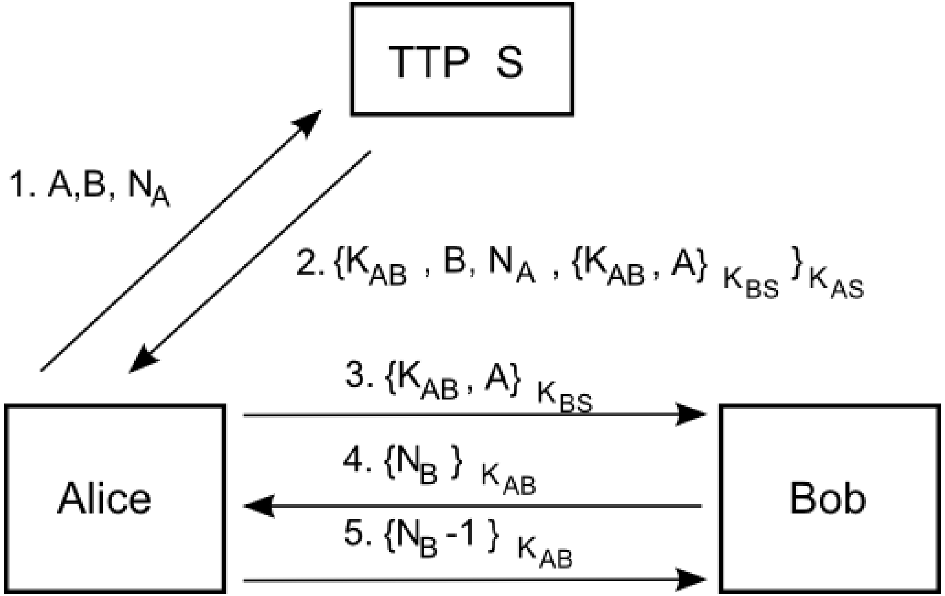
Key distribution


TTP S
1. A,B
2. $\{K_{AB}\}_{K_{AS}}, \{K_{AB}\}_{K_{BS}}$
Alice
3. $\{K_{AB}\}_{K_{BS}}, A$
Bob

Key distribution


TTP S
1. A,B
2. {K_{AB} , B} K_{AS} , {K_{AB} , A} K_{BS}
3. {K_{AB} , A} K_{BS}
Alice
Bob

Key distribution



TTP S

1. A,B, N_A

2. $\{K_{AB}, B, N_A, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$

3. $\{K_{AB}, A\}_{K_{BS}}$

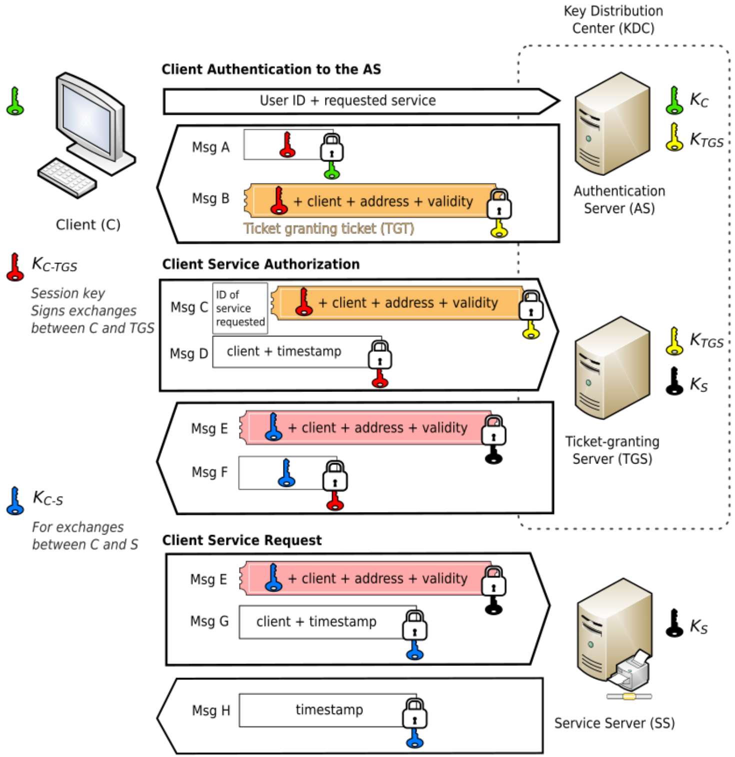
4. $\{N_B\}_{K_{AB}}$

5. $\{N_B - 1\}_{K_{AB}}$

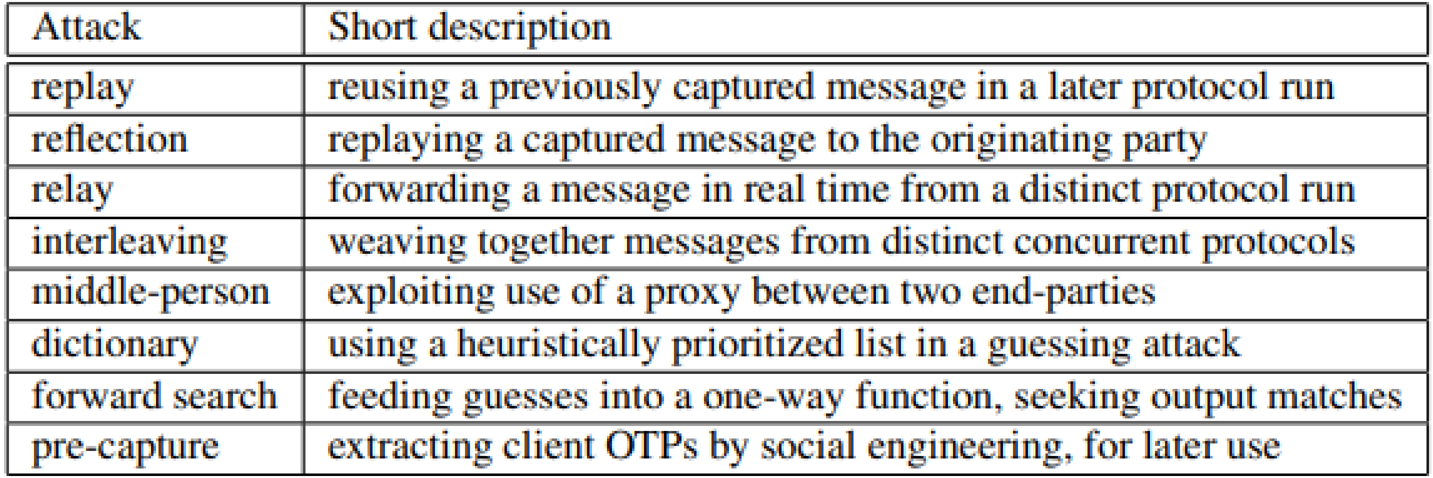
Alice

Bob

Kerberos


Key Distribution Center (KDC)
Client Authentication to the AS
User ID + requested service
Msg A
Msg B
+ client + address + validity
Ticket granting ticket (TGT)
Client (C)
Authentication Server (AS)
K_C
K_{TGS}
$K_{C\text{-}TGS}$
Session key
Signs exchanges between C and TGS
Client Service Authorization
Msg C
ID of service requested
+ client + address + validity
Msg D
client + timestamp
K_{TGS}
K_S
Msg E
+ client + address + validity
Msg F
Ticket-granting Server (TGS)
$K_{C\text{-}S}$
For exchanges between C and S
Client Service Request
Msg E
+ client + address + validity
Msg G
client + timestamp
Msg H
timestamp
K_S
Service Server (SS)

More key management risks

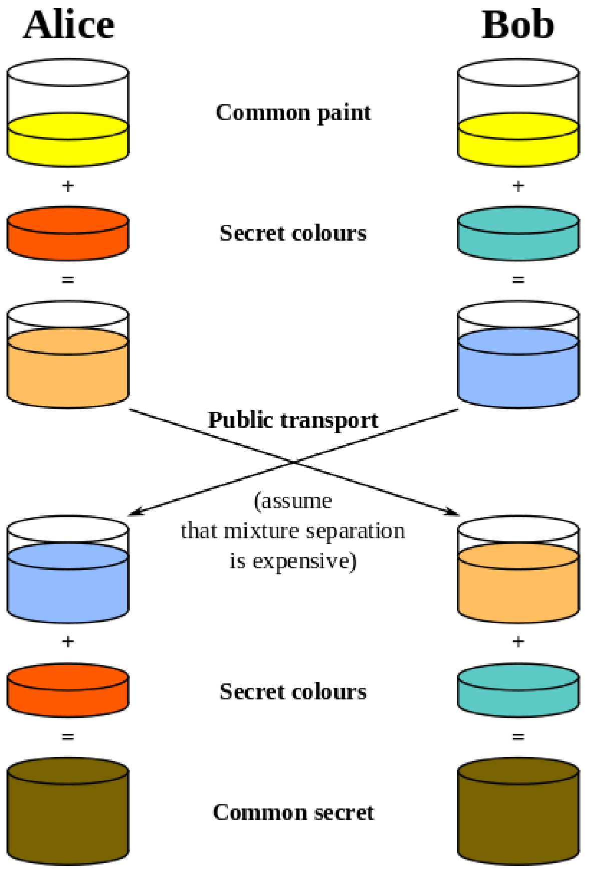


Attack	Short description
replay	reusing a previously captured message in a later protocol run
reflection	replaying a captured message to the originating party
relay	forwarding a message in real time from a distinct protocol run
interleaving	weaving together messages from distinct concurrent protocols
middle-person	exploiting use of a proxy between two end-parties
dictionary	using a heuristically prioritized list in a guessing attack
forward search	feeding guesses into a one-way function, seeking output matches
pre-capture	extracting client OTPs by social engineering, for later use

Key agreement

Basic idea

If you wanted to exchange secret paints



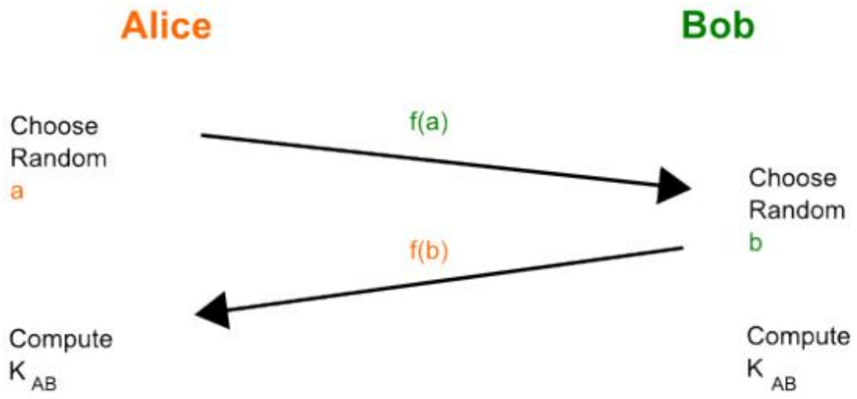
Basic idea

Choose a function f such that

$$f(a,f(b)) = f(b,f(a))$$

And

$$f^{-1}(x) \text{ is hard}$$



Solution by Diffie-Hellman, 1976

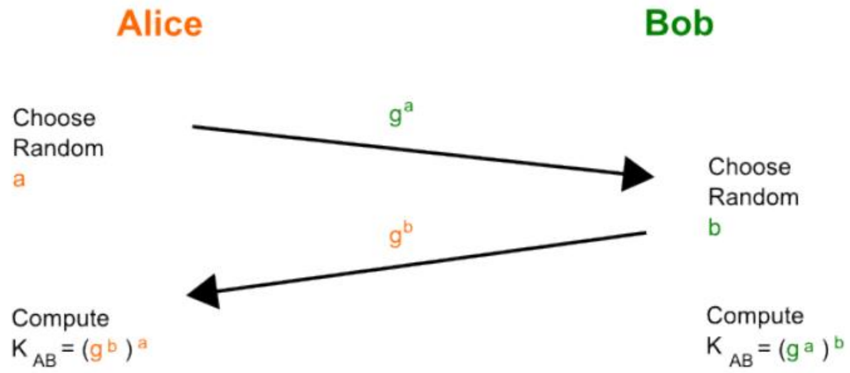
$f(x) = g^x \bmod p$

Given g^a, find x so $g^x=g^a$

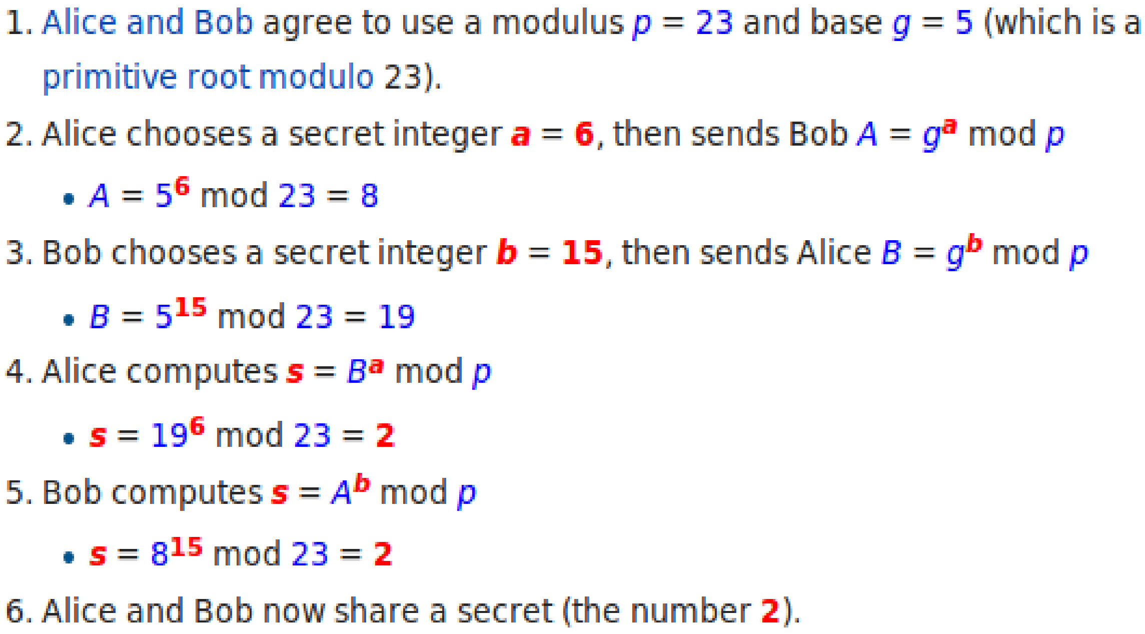
Discrete logarithm problem

Given g^a and g^b, find g^{ab}

Computational Diffie-Hellman assumption



Diffie-Hellman: toy example

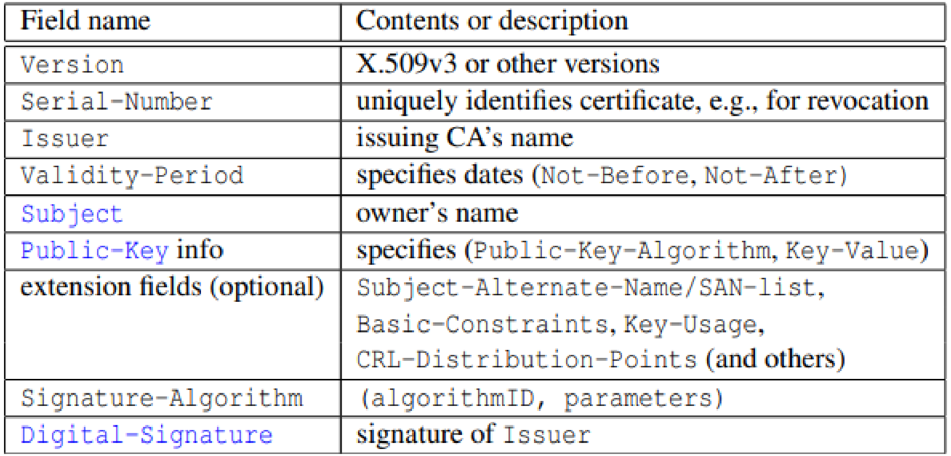


1. Alice and Bob agree to use a modulus $p = 23$ and base $g = 5$ (which is a primitive root modulo 23).
2. Alice chooses a secret integer $\mathbf{a} = \mathbf{6}$, then sends Bob $A = g^{\mathbf{a}} \bmod p$
   - $A = 5^{\mathbf{6}} \bmod 23 = 8$
3. Bob chooses a secret integer $\mathbf{b} = \mathbf{15}$, then sends Alice $B = g^{\mathbf{b}} \bmod p$
   - $B = 5^{\mathbf{15}} \bmod 23 = 19$
4. Alice computes $\mathbf{s} = B^{\mathbf{a}} \bmod p$
   - $\mathbf{s} = 19^{\mathbf{6}} \bmod 23 = \mathbf{2}$
5. Bob computes $\mathbf{s} = A^{\mathbf{b}} \bmod p$
   - $\mathbf{s} = 8^{\mathbf{15}} \bmod 23 = \mathbf{2}$
6. Alice and Bob now share a secret (the number **2**).

Is *e* really yours?

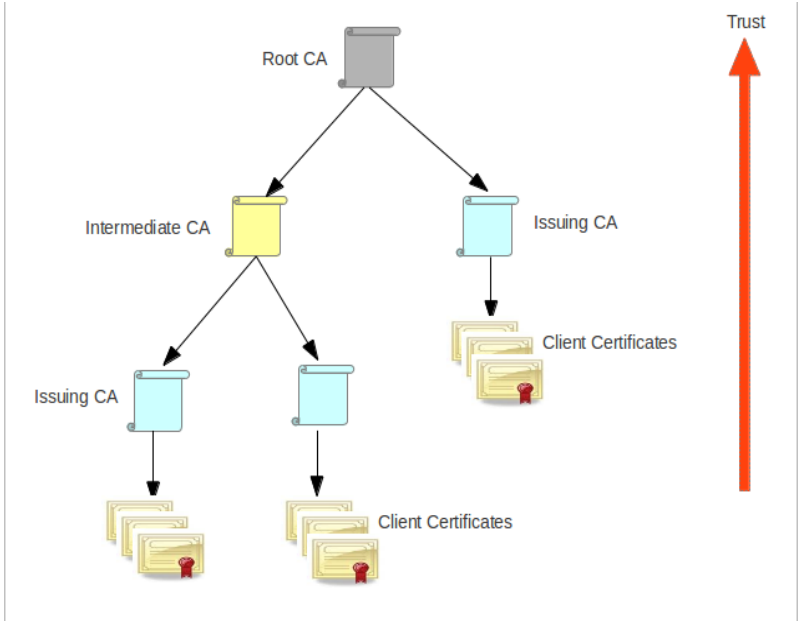
Public-key infrastructure (PKI)

A system for the creation, storage, and distribution of **digital certificates** which are used to verify that a particular public key belongs to a certain entity

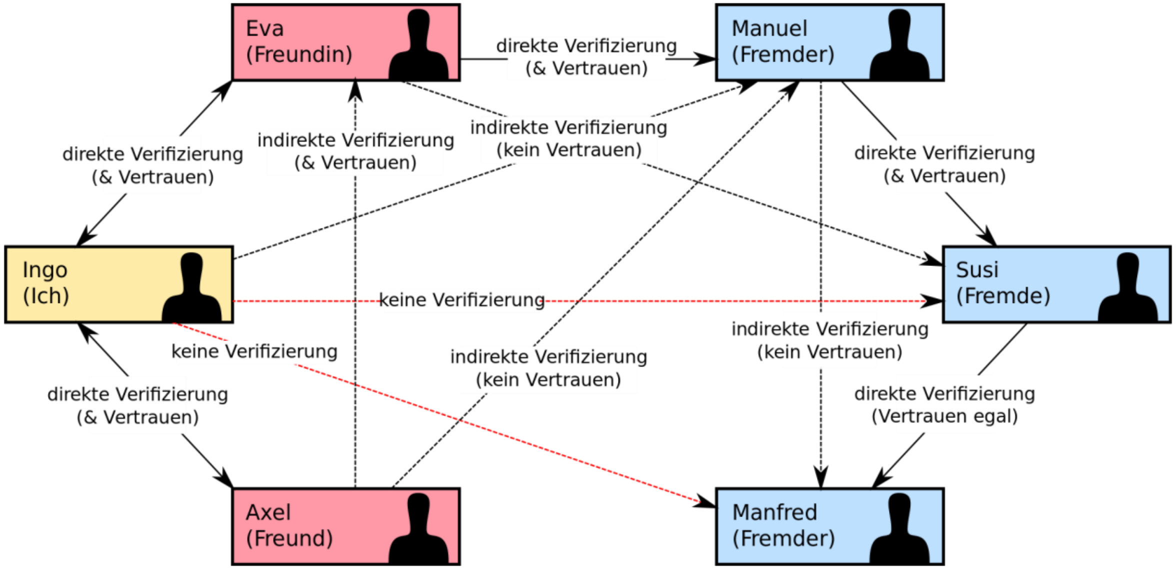


Field name	Contents or description
Version	X.509v3 or other versions
Serial-Number	uniquely identifies certificate, e.g., for revocation
Issuer	issuing CA's name
Validity-Period	specifies dates (Not-Before, Not-After)
Subject	owner's name
Public-Key info	specifies (Public-Key-Algorithm, Key-Value)
extension fields (optional)	Subject-Alternate-Name/SAN-list, Basic-Constraints, Key-Usage, CRL-Distribution-Points (and others)
Signature-Algorithm	(algorithmID, parameters)
Digital-Signature	signature of Issuer

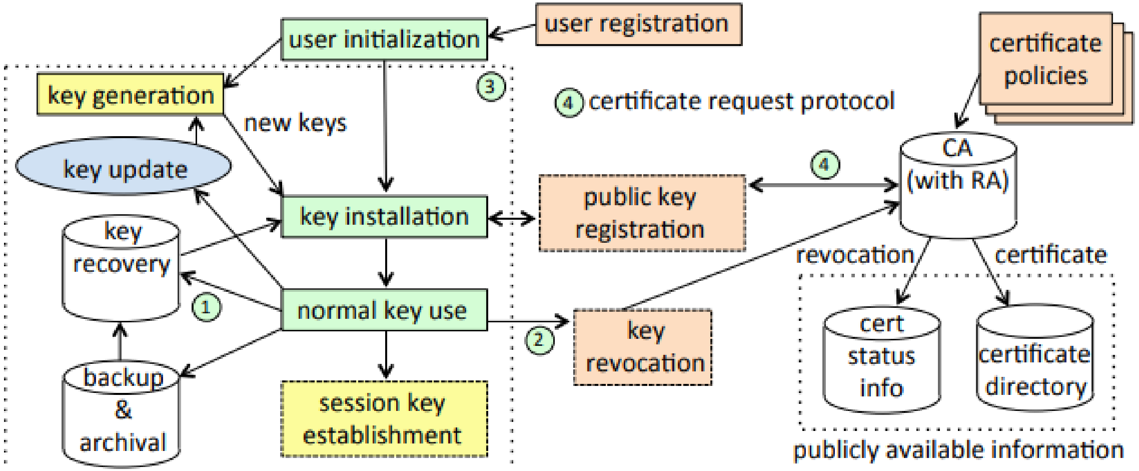
Types of PKI: CA model



Types of PKI: Web of trust



PKI components and lifecycle

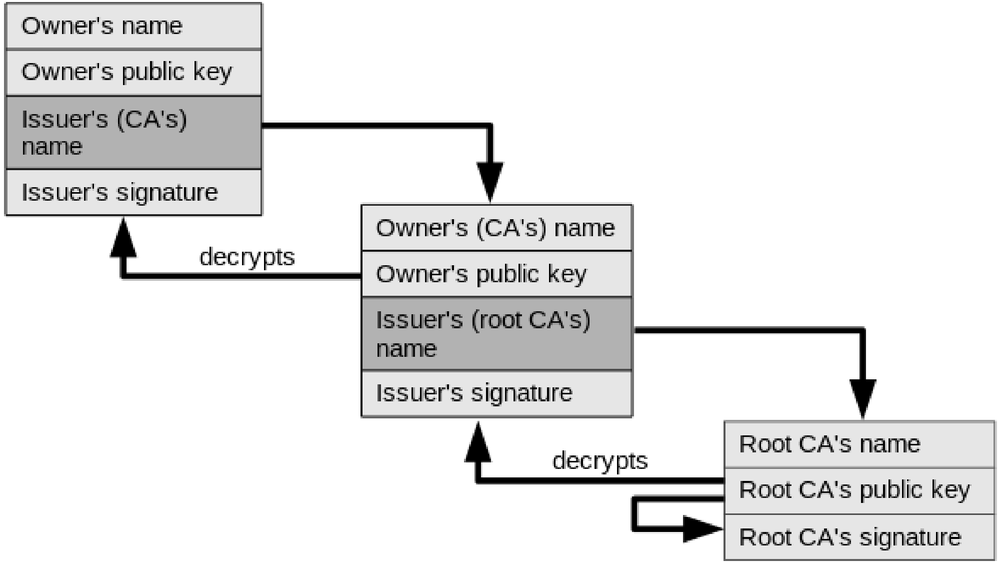


① key loss without compromise ② key compromise or early termination ③ private key management

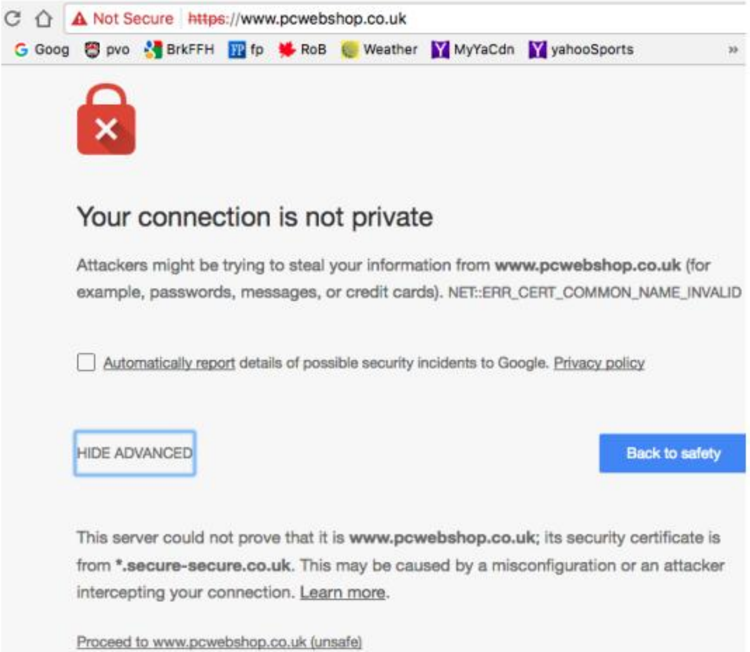
Certificate validation

1. Not expired
2. Not revoked
3. Its signature verifies
4. Stated use matches intended use
5. Signed by CA that is trusted OR chain that leads to a CA that is trusted

Chain of trust



Browsing untrusted sites



Your connection is not private

Attackers might be trying to steal your information from **www.pcwebshop.co.uk** (for example, passwords, messages, or credit cards). NET::ERR_CERT_COMMON_NAME_INVALID

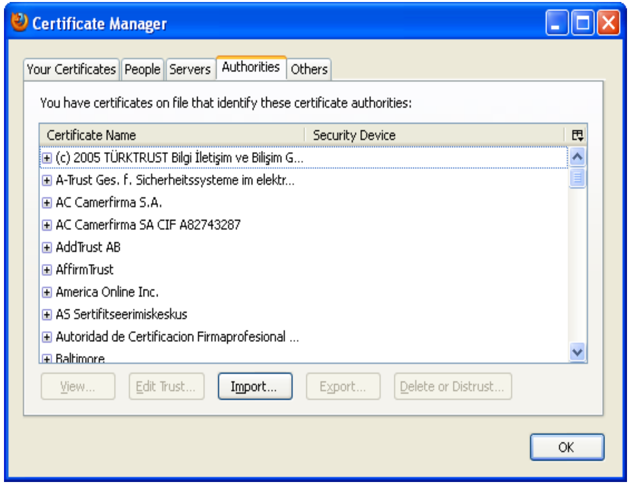
☐ Automatically report details of possible security incidents to Google. Privacy policy

This server could not prove that it is **www.pcwebshop.co.uk**; its security certificate is from ***.secure-secure.co.uk**. This may be caused by a misconfiguration or an attacker intercepting your connection. Learn more.

Proceed to www.pcwebshop.co.uk (unsafe)

Trust in browsers

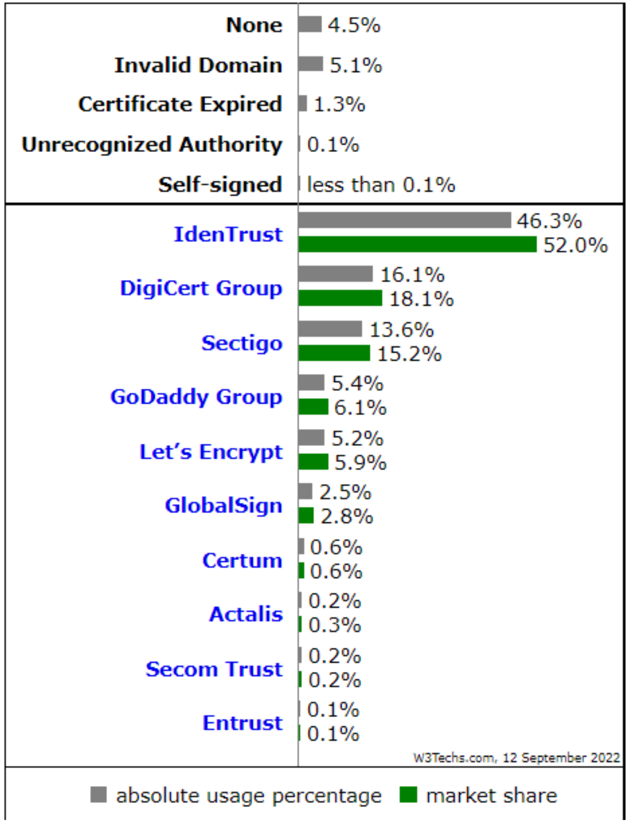
Browsers come pre-configured with a set of root CAs. Do you trust all these CAs (to authenticate properly, to avoid/inform of breaches)?



You have certificates on file that identify these certificate authorities:

- (c) 2005 TÜRKTRUST Bilgi İletişim ve Bilişim G...
- A-Trust Ges. f. Sicherheitssysteme im elektr...
- AC Camerfirma S.A.
- AC Camerfirma SA CIF A82743287
- AddTrust AB
- AffirmTrust
- America Online Inc.
- AS Sertifitseerimiskeskus
- Autoridad de Certificacion Firmaprofesional ...
- Baltimore

CA providers



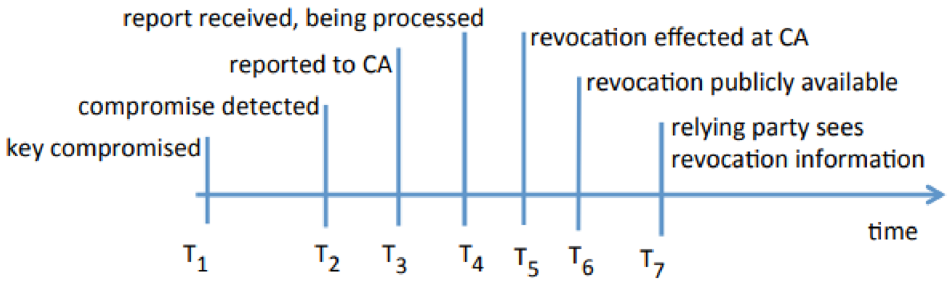
Revocation of certificates

Certificate revocation list (CRL):

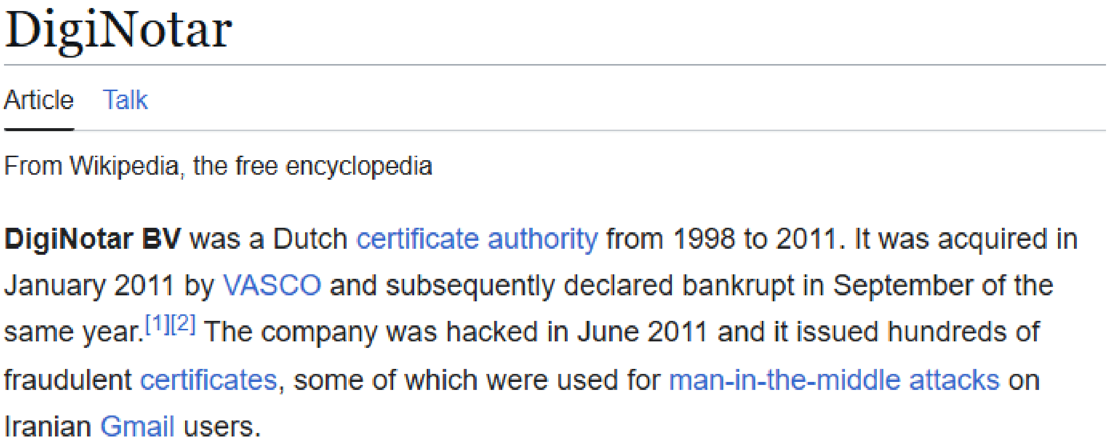
A list of (serial numbers for) certificates that have been revoked, and therefore, entities presenting those (revoked) certificates should no longer be trusted

Online Certificate Status Protocol (OCSP):

Protocol used for obtaining the revocation status of an X.509 digital certificate


report received, being processed
revocation effected at CA
reported to CA
revocation publicly available
compromise detected
relying party sees revocation information
key compromised
time
T_1
T_2
T_3
T_4
T_5
T_6
T_7

CA breach

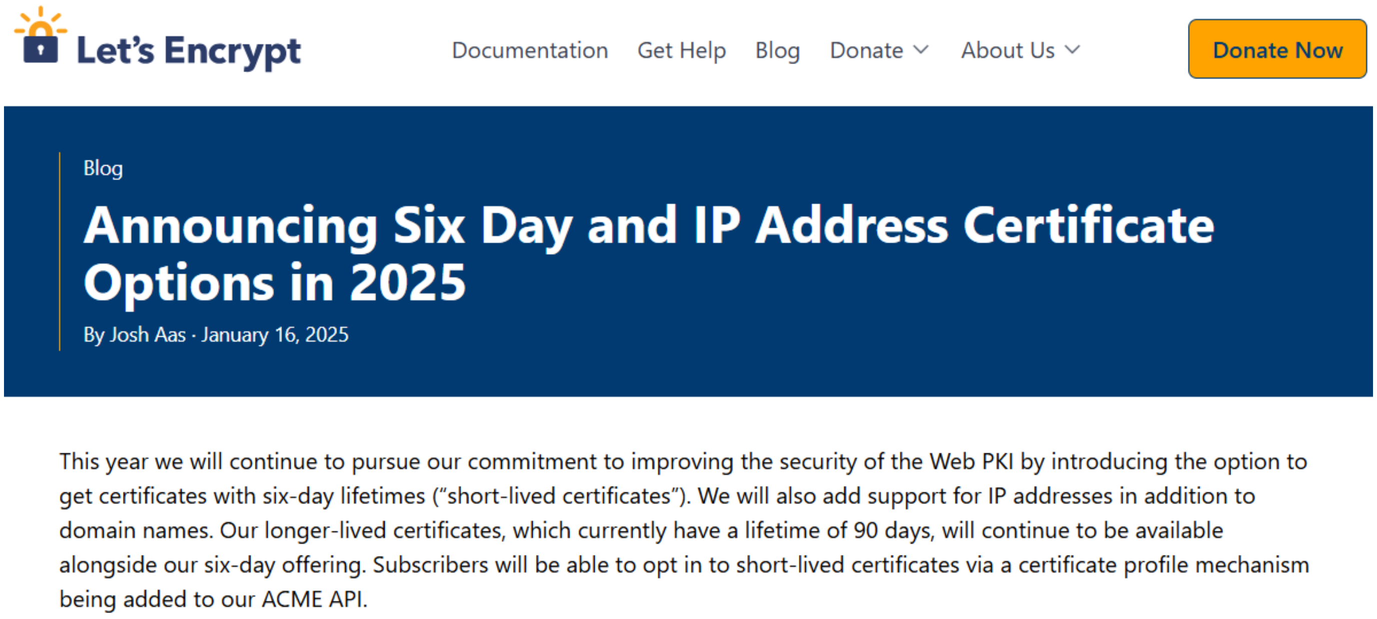


DigiNotar

From Wikipedia, the free encyclopedia

DigiNotar BV was a Dutch certificate authority from 1998 to 2011. It was acquired in January 2011 by VASCO and subsequently declared bankrupt in September of the same year.[1][2] The company was hacked in June 2011 and it issued hundreds of fraudulent certificates, some of which were used for man-in-the-middle attacks on Iranian Gmail users.

Short-lived certificates



Blog

Announcing Six Day and IP Address Certificate Options in 2025

By Josh Aas · January 16, 2025

This year we will continue to pursue our commitment to improving the security of the Web PKI by introducing the option to get certificates with six-day lifetimes ("short-lived certificates"). We will also add support for IP addresses in addition to domain names. Our longer-lived certificates, which currently have a lifetime of 90 days, will continue to be available alongside our six-day offering. Subscribers will be able to opt in to short-lived certificates via a certificate profile mechanism being added to our ACME API.