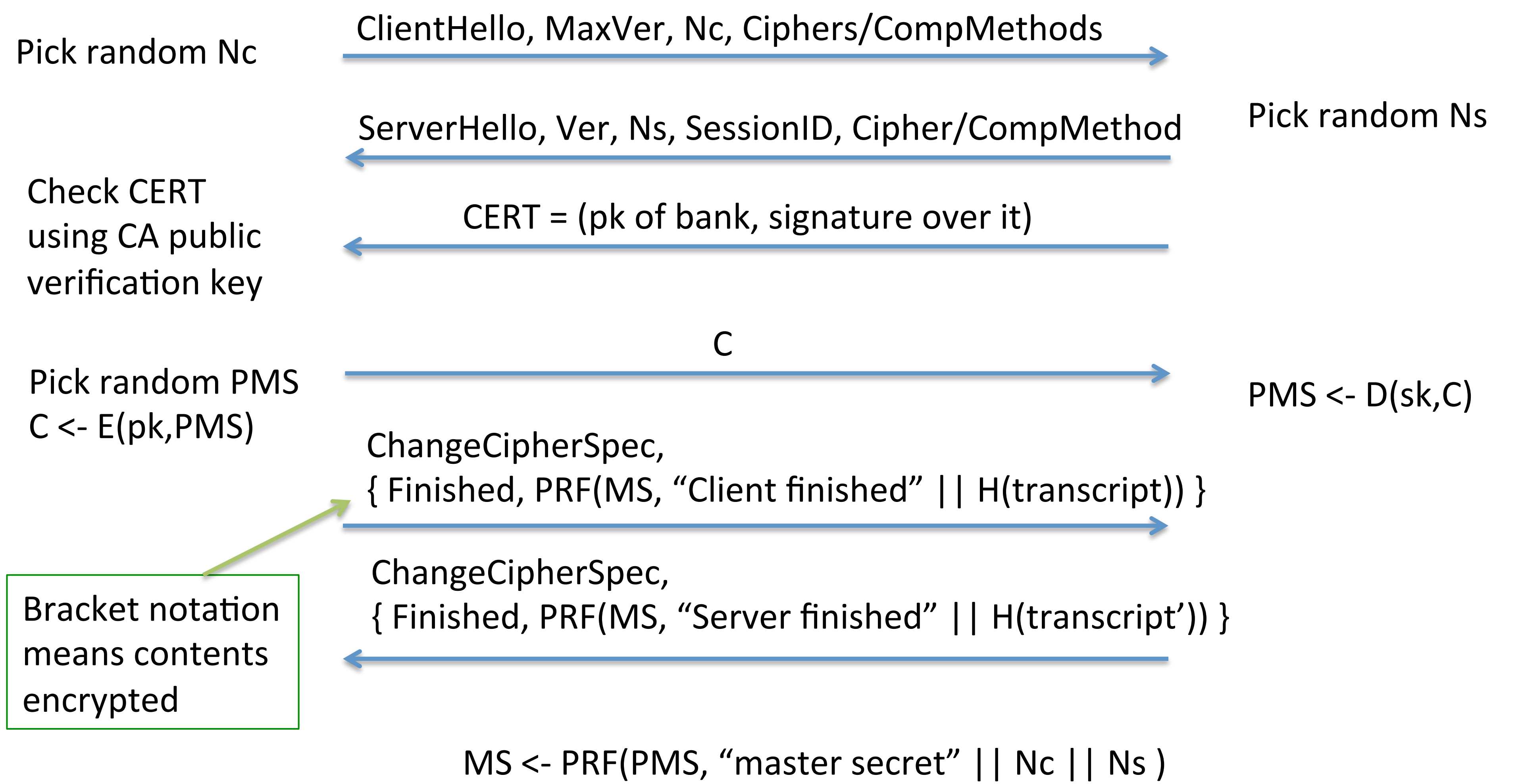
Today in Cryptography (5830)

Digital signatures
RSA signatures and full domain hash
Schnorr signatures, DSA
PKI

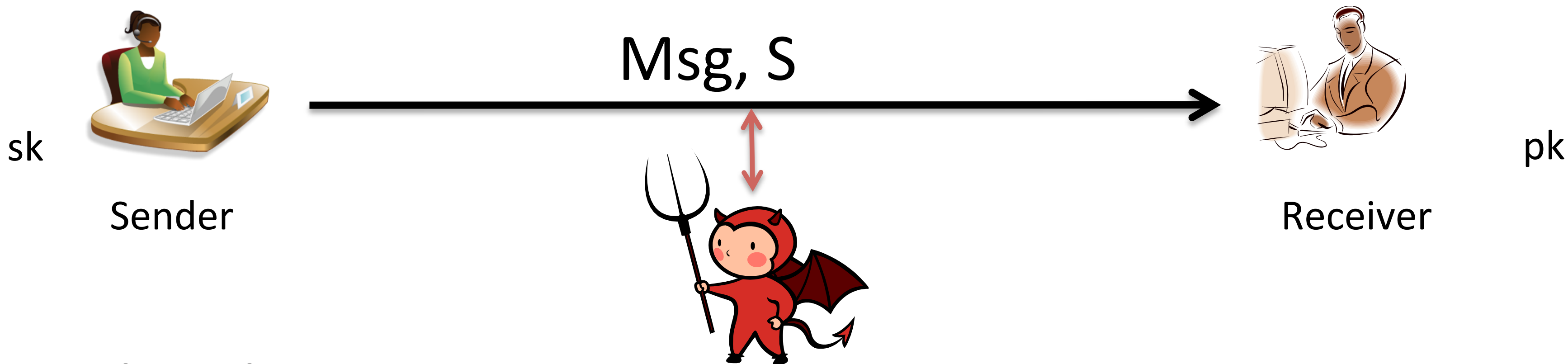
TLS handshake for RSA transport

Client

Server



Digital signatures



Two algorithms:

(1) Key generation outputs (pk,sk)

(2) Sign (sk,Msg) outputs a signature S (may be randomized)

(3) Verify(pk,Msg,S) outputs 0/1 (invalid / valid)

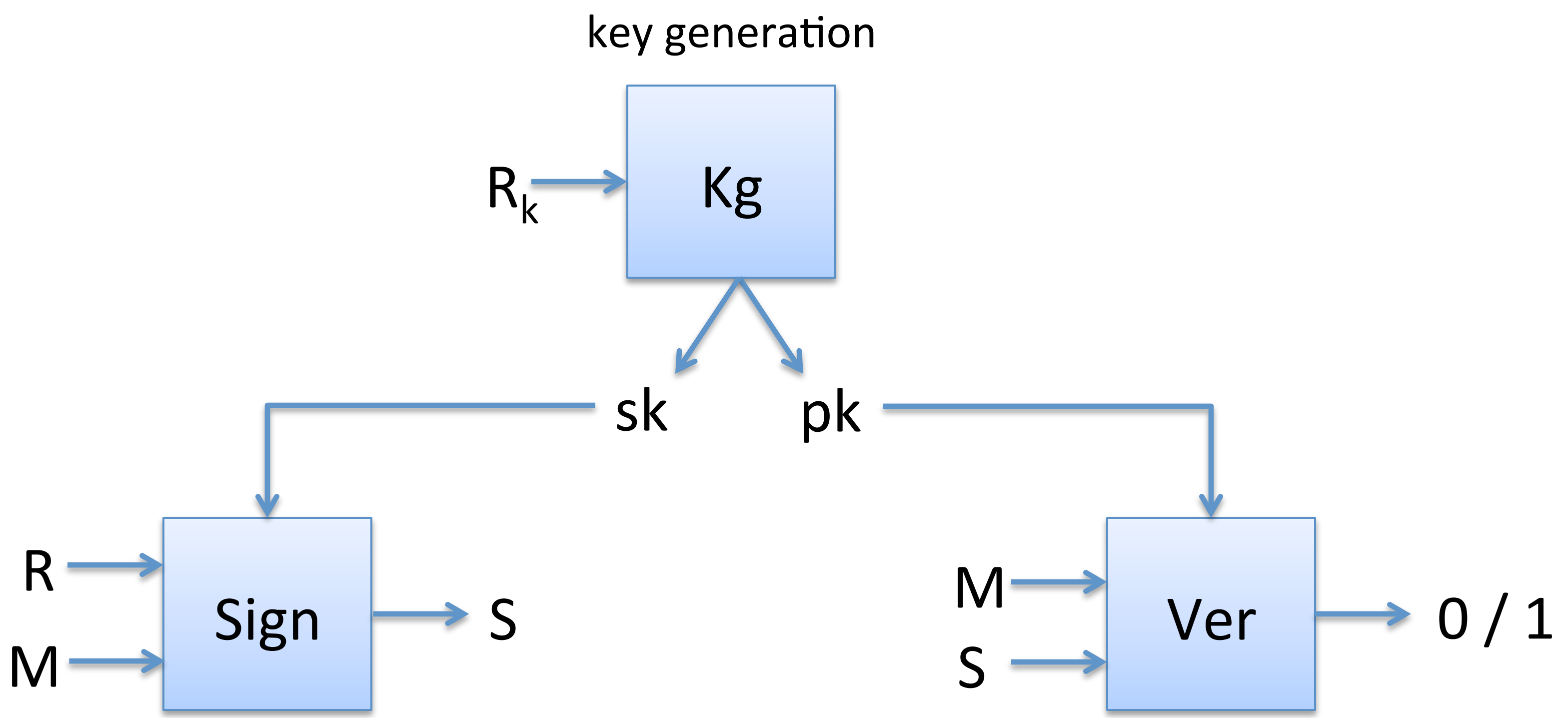
Correctness: Verify(pk,Msg,Sign(sk,Msg)) = 1 always

Security: No computationally efficient attacker can forge signatures for a new message even when attacker gets

$$(\text{Msg}_1, S_1), (\text{Msg}_2, S_2), \dots, (\text{Msg}_q, S_q)$$

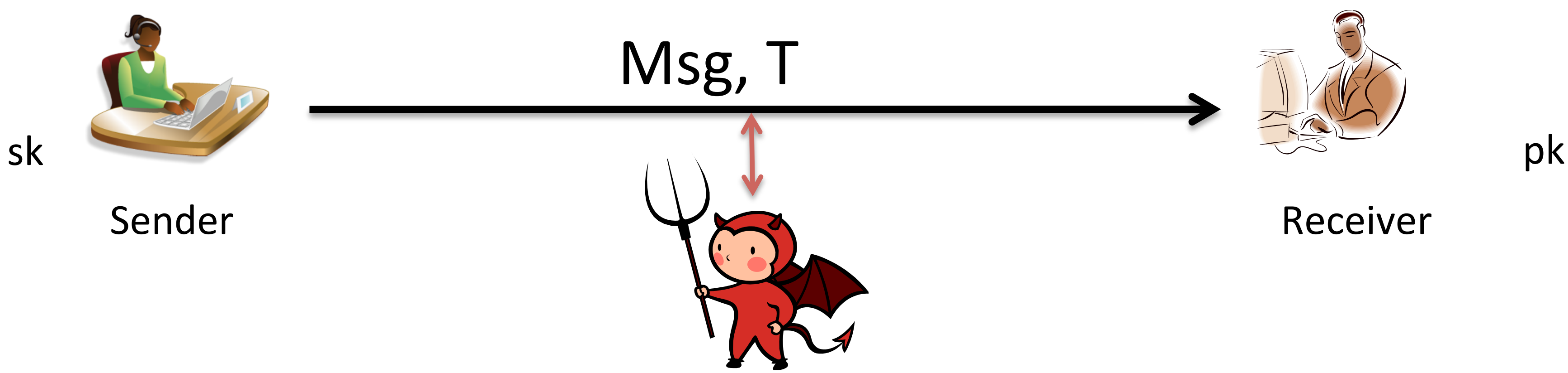
for messages of his choosing and reasonably large q.

Digital signatures



Anyone with public key can verify a signature
Only holder of secret key should be able to generate a signature

Digital signatures



“Raw” RSA as a signature scheme:

Key generation gives (N,e) , (N,d)
Sign((N,d),M) = M^d mod N
Verify((N,e),M,S) checks if S^e mod N = M

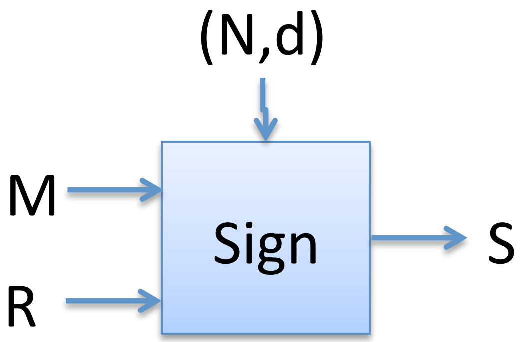
Secure? No!

PKCS #1 RSA signing

Kg outputs (N,e),(N,d) where $|N|_8 = n$

Let $B = \{0,1\}^8 / \{00\}$ be set of all bytes except 00

Want to encrypt messages of length $|M|_8 = m$

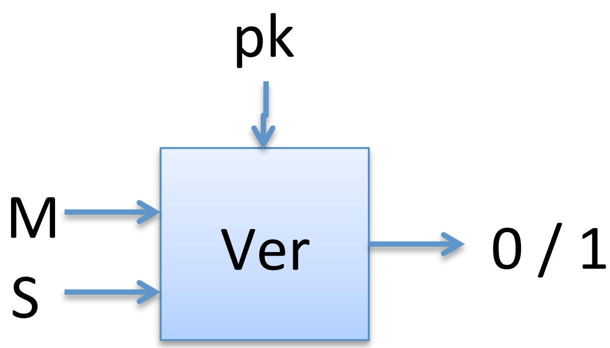


<u>Sign((N,d), M , R)</u>

pad = first n - m - 2 bytes from R that are in B

Y = 00 || 01 || pad || 00 || H(M)

Return Y^d mod N



<u>Verify((N,e), M, S)</u>

$Y = C^e$ mod N ; aa||bb||w = Y

If (aa $\neq$ 00) or (bb $\neq$ 01) or (00 $\notin$ w)

Return error

pad || 00 || h = w

Return H(M) = h

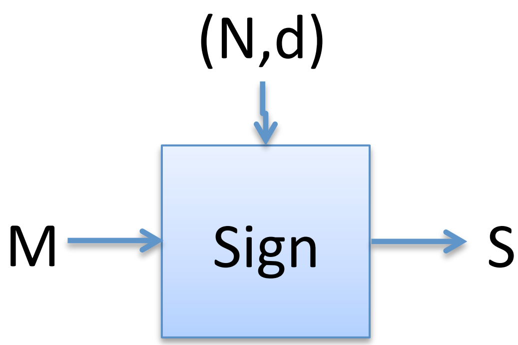
Digital signature security

- Padding oracle attacks that work against RSA PKCS#1 v1.5 decryption work against similar implementations of signing

Full Domain Hash RSA

Kg outputs pk = (N,e) , sk = (N,d)

H is a hash function

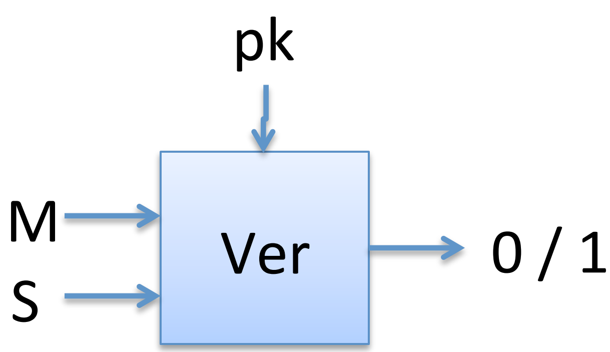


<u>Sign((N,d), M)</u>

X = 00 || H(1||M) || … || H(k||M)

S = X^d mod N

Return S



<u>Ver((N,e), M, S)</u>

X = S^e mod N

X’ = 00 || H(1||M) || … || H(k||M)

If X = X’ then

    Return 1

Return 0

Schnorr signatures

Choose prime q and we'll work in multiplicative group $\mathbf{Z}_q^*$
sk = k chosen in $\mathbf{Z}_q$ pk = g^k

<u>Sign(k, M)</u>
$R = g^r$; $e = H(M \,||\, R)$; $s = r - xe$
Return (s,e)

<u>Ver(pk = g^k, M, (s,e))</u>
$R_v = g^s * pk^e$; $e_v = H(M \,||\, R_v)$
If $e_v = e$ then Return 1
Return 0

DSA (digital signature algorithm)

Choose prime q and p s.t. p-1 | q . Set $g = h^{(p-1)/q} \bmod p$

$sk = k$ chosen in $\mathbf{Z}_q$ $pk = g^k$

<u>Sign(k, M)</u>
$r \leftarrow\$ \mathbf{Z}_q$ until $R = (g^r \bmod p) \bmod q \neq 0$
$s \leftarrow k^{-1}(H(M) + k R) \bmod q$ (start over if s = 0)
Return (R,s)

<u>Ver(pk = g^k, M, (R,s))</u>
If R,s not in $\mathbf{Z}_q$
$w \leftarrow s^{-1} \bmod q$; $u1 = H(m) * w \bmod q$
$u2 = R*w \bmod q$; $v = (g^{u1}\, pk^{u2} \bmod p) \bmod q$
If v = R then Return 1
Return 0

Client

TLS handshake for RSA transport

Server

Pick random Nc

ClientHello, MaxVer, Nc, Ciphers/CompMethods

Pick random Ns

ServerHello, Ver, Ns, SessionID, Cipher/CompMethod

Check CERT using CA public verification key

CERT = (pk of server, signature over it)

C

Pick random PMS
C <- E(pk,PMS)

PMS <- D(sk,C)

ChangeCipherSpec,
{ Finished, PRF(MS, “Client finished” || H(transcript)) }

Bracket notation means contents encrypted

ChangeCipherSpec,
{ Finished, PRF(MS, “Server finished” || H(transcript’)) }

MS <- PRF(PMS, “master secret” || Nc || Ns)

Certificate Authorities and Public-key Infrastructure

(pk,sk)

$M = (pk',data)$
$S = Sign(sk,M)$

pk

Give me a certificate for pk', please

S

http://amazon.com

pk', data, S

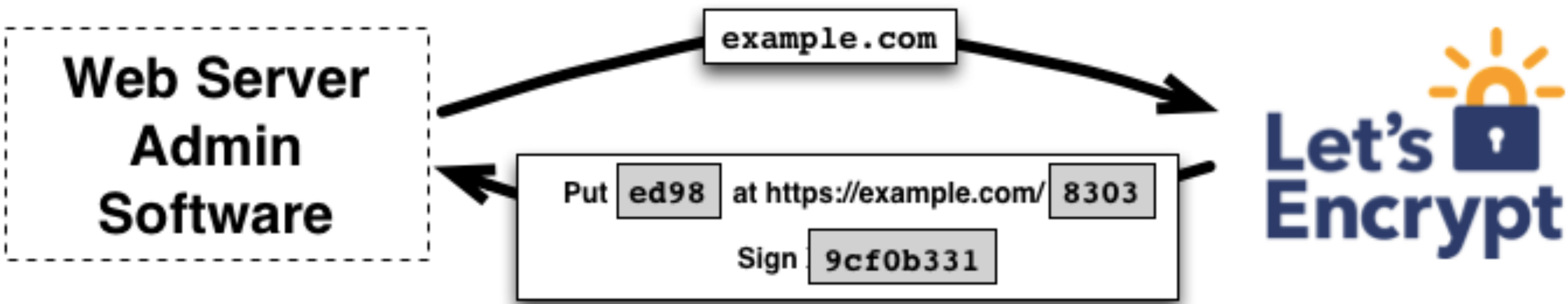
(pk',sk')

$M = (pk',data)$
If $Ver(pk,M,S)$ then
trust pk'

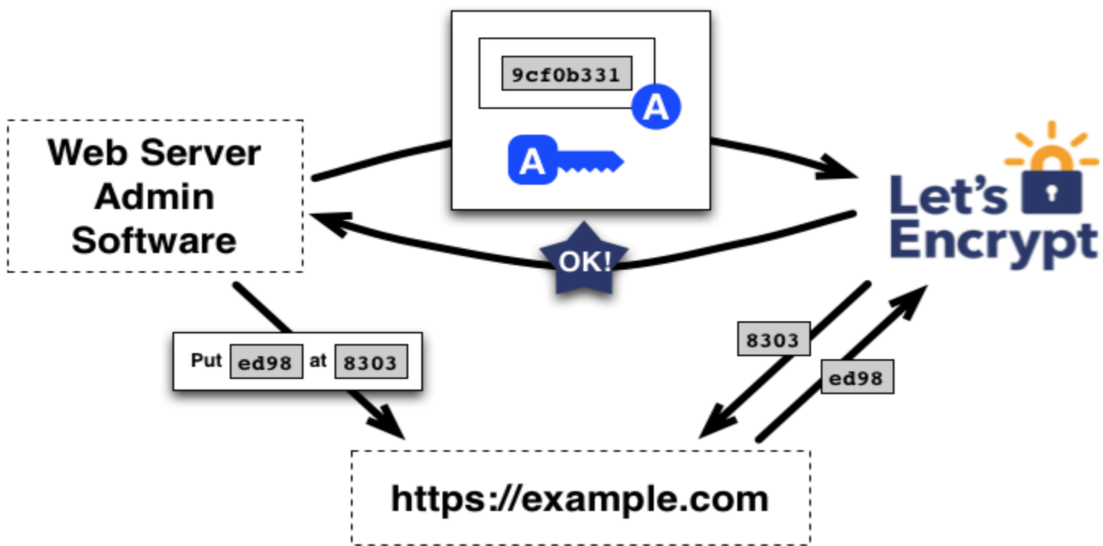
This prevents man-in-the-middle (MitM) attacks

```
Certificate:
   Data:
       Version: 1 (0x0)
       Serial Number: 7829 (0x1e95)
       Signature Algorithm: md5WithRSAEncryption
       Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
               OU=Certification Services Division,
               CN=Thawte Server CA/emailAddress=server-certs@thawte.com
       Validity
           Not Before: Jul  9 16:04:02 1998 GMT
           Not After : Jul  9 16:04:02 1999 GMT
       Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
                OU=FreeSoft, CN=www.freesoft.org/emailAddress=baccala@freesoft.org
       Subject Public Key Info:
           Public Key Algorithm: rsaEncryption
           RSA Public Key: (1024 bit)
               Modulus (1024 bit):
                   00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
                   33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
                   66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:
                   70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
                   16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:
                   c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
                   8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:
                   d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
                   e8:35:1c:9e:27:52:7e:41:8f
               Exponent: 65537 (0x10001)
   Signature Algorithm: md5WithRSAEncryption
       93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
       92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:
       ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:
       d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:
       0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:
       5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:
       8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:
       68:9f
```

Free CAs



Web Server Admin Software
example.com
Let's Encrypt
Put ed98 at https://example.com/ 8303
Sign 9cf0b331



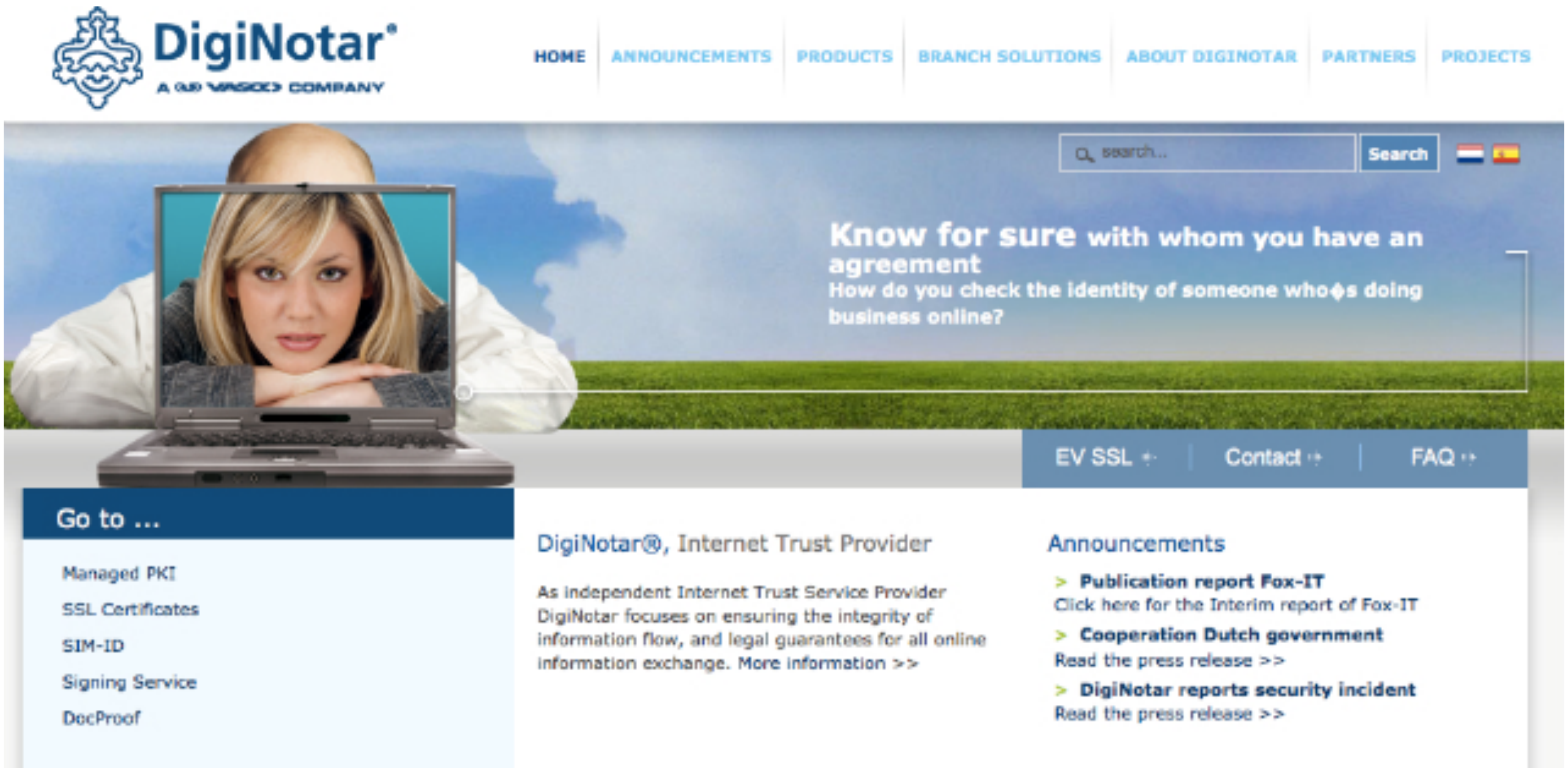
9cf0b331
A
A
Web Server Admin Software
Let's Encrypt
OK!
Put ed98 at 8303
8303
ed98
https://example.com

Revocation

- Certificates must often be revoked
  - Short expirations
  - CRLs (Certificate revocation lists)
  - OCSP (online certificate status protocol)

The Web PKI Ecosystem

- http://conferences.sigcomm.org/imc/2013/papers/imc257-durumericAemb.pdf
- ~1800 CAs that can sign *any* domain controlled by 683 organizations


DigiNotar®
HOME
ANNOUNCEMENTS
PRODUCTS
BRANCH SOLUTIONS
ABOUT DIGINOTAR
PARTNERS
PROJECTS
search...
Search
Know for sure with whom you have an agreement
How do you check the identity of someone who�s doing business online?
EV SSL
Contact
FAQ
Go to ...
Managed PKI
SSL Certificates
SIM-ID
Signing Service
DocProof
DigiNotar®, Internet Trust Provider
As independent Internet Trust Service Provider DigiNotar focuses on ensuring the integrity of information flow, and legal guarantees for all online information exchange. More information >>
Announcements
> Publication report Fox-IT
Click here for the Interim report of Fox-IT
> Cooperation Dutch government
Read the press release >>
> DigiNotar reports security incident
Read the press release >>