Cryptography 4 – Addendum: (EC)DSA Signatures

Maria Eichlseder

Information Security – WT 2019/20

(EC)DSA Signatures

- Security relies on Diffie–Hellman Problem (DHP) / Discrete Log Problem (DLP)
- The algorithm involves a long-term keypair $x, y = \alpha^x$ like Diffie–Hellman, but also an ephemeral (temporary) keypair k, α^k.
  - This makes the signature scheme non-deterministic
  - k must be unpredictable and secret and **must not be reused**!
- There are several related schemes:
  - ElGamal signature, the original & the simplest
  - DSA signature, a more efficient version (some numbers are smaller)
  - ECDSA, Elliptic-Curve version of DSA where some multiplications are replaced by point additions and exponentiations by scalar multiplications

DSA (Digital Signature Algorithm, FIPS 186-4)

Key generation

Choose a prime p, prime q (q divides $p-1$), and $\alpha \in \mathbb{Z}_p^*$ with $\alpha^q \equiv 1 \pmod{p}$:

$$\text{private key} = x \in \{2, \ldots, p-1\} \qquad \text{public key} = y = \alpha^x \pmod{p}$$

Sign

Hash $h = \text{SHA-2}(M)$. Choose ephemeral key $k \in \{2, \ldots, q-1\}$ co-prime to q:

$$S = (r, s): \qquad r = (\alpha^k \bmod p) \bmod q, \qquad s = (h + xr) \cdot k^{-1} \pmod{q}$$

Verify

Compute $\quad w = s^{-1} \bmod q, \quad$ verify that $\quad r \stackrel{?}{=} (\alpha^{h \cdot w} \cdot y^{r \cdot w} \bmod p) \bmod q$

DSA signature: Security

- Ephemeral key k must not be reused!

  Suppose $h_1 = \text{SHA-2}(M_1)$, $h_2 = \text{SHA-2}(M_2)$ were signed with the same k, then by rewriting the definition of s (s_1 for M_1 and s_2 for M_2), we get:

  $$\begin{aligned} h_1 &= k \cdot s_1 - x \cdot r \\ h_2 &= k \cdot s_2 - x \cdot r \\ h_1 - h_2 &= k \cdot (s_1 - s_2) \end{aligned}$$

  We can recover $k = (h_1 - h_2) \cdot (s_1 - s_2)^{-1}$ and thus also the private key x!

  If k is not exactly the same, but somehow related, similar tricks are possible.

  Real-world examples: Sony PS3 disaster 2010, Debian PRNG bug 2008

DSA signature: Security

- Prime bitsize recommendations (NIST SP 800-57):

Security in bits	80	112	128	192	256
Size of p in bits	1024	2048	3072	7680	15360
Size of q in bits	160	224	256	384	512

- Advantages and disadvantages of DSA (compared to RSA signatures)
  - ⊕ some operations mod q (smaller)
  - ⊕ shorter signature
  - ⊖ cannot be (easily) used for encryption (crypto export restrictions!)
  - ⊖ verification of signature slower than signing
- Same problem as ElGamal with reuse of k

EC ElGamal/DSA signatures (ECDSA, simplified)

Classic ElGamal/DSA signature

Public: prime p, number $\alpha \in \mathbb{Z}_p^*$ of prime order q

Alice's keypair: $x \in \mathbb{N}$, $y = \alpha^x$

Hash $H \in \mathbb{N}$ of message M

Signing of M by Alice:

Alice picks $k \in \mathbb{N}$

$$r = [\alpha^k]_{\text{mod } q}$$
$$s = [k^{-1} \cdot (H + xr)]_{\text{mod } q}$$

$\xrightarrow{(r,s)}$ Bob verifies

$$r \stackrel{?}{=} [\alpha^{Hs^{-1}} y^{rs^{-1}}]_q$$

EC ElGamal/ECDSA signature

Public: curve E, base point $\alpha \in E$ of prime order q

Alice's keypair: $x \in \mathbb{N}$, $y = x \cdot \alpha$

Hash $H \in \mathbb{N}$ of message M

Signing of M by Alice:

Alice picks $k \in \mathbb{N}$

$$r = [k \cdot \alpha]_{x\text{-coord mod } q}$$
$$s = [k^{-1} \cdot (H + xr)]_{\text{mod } q}$$

$\xrightarrow{(r,s)}$ Bob verifies

$$r \stackrel{?}{=} [Hs^{-1} \cdot \alpha + rs^{-1} \cdot y]_{x,q}$$