Oct22

Wednesday, October 21, 2020 10:02 PM

① Brown Bag Lunch:

on Zoom on Oct 27

Register by Friday

(info sesh for those interested in Careers in mathematics).

② Undergraduate Number Theory Conference

Nov 14-15

- 4 Famous Speakers.
- Problem Sessions
- Panels

Digital Signatures

Abstract Setup

Sam (K^{priv}, K^{pub}) D document ^{pub}

Uses K^{priv} to produce D^{sig} from D

Victor (the verifyer) use K^{pub} to verify that D^{sig} is the output when D is signed by $K^{priv.}$

Subtlety

Want Check that $(K^{priv}, D) \leftrightarrow D^{sig}$ match w/o being able to produce D^{sig} from D (only Sam can do that).

↑ Forge the document.

Important

- * Eve cannot produce D^{sig} from D
- * Anyone can check if $D^{sig} \leftrightarrow D$ are a valid pair.

Example

If Eve convinces me that $\hat{K}^{pub}$ is the public verification key then she can feed me forged docs w/o ever forging Sam's signature.

↑ Solution More Public = Better

RSA Digital Signatures

Setup is pretty much same as RSA.

Note Replace Eve not being able to read message w/ Eve not being ~~be~~ able to forge document. (Sign as Sam).

3 steps
① Key Creation
② Signing
③ Verifying Signatures.

① Key Creation

Samantha choose 2 secret primes p, q.

Choose e a verification exponent

Computes $d = e^{-1} \bmod (p-1)(q-1)$ signing exponent

Computes $N = pq$

Publishes $K^{pub} = [N, e]$

public
secret

② Signing Documents

Document $D \in \mathbb{Z}/N\mathbb{Z}$

Samanth. signs $D^{sig} \equiv D^{d} \bmod N$

③ Verification

Victor knows (D, D^{sig})

Victor computes $(D^{sig})^{e} \bmod N$

Verifies that he recovered D.

Idea: backwards RSA

Sam compute $\sqrt[e]{D}$. ← only Sam knows how to do this.

To verify, Victor raises to e'th power.

If victor recovers D, then Sam must have signed the doc b/c only Sam can compute e'th roots

↑ same reason why we can securely sent messages to Sam by raising them to d.

Proof:

want to show that verification recognizes sigs.

$$(D^{sig})^e \underset{\substack{\uparrow \\ \text{b/c Sam signed}}}{=} (D^d)^e = D^{de} = D^{1+k(p-1)(q-1)}$$

$$\underset{\text{Euler}}{\equiv} D \mod N$$

Attacks If Eve can factor $N=pq$.

She can compute d, & then forge Sam's signature: raising to d.

Recall

Eve tricks Alice into decoding a message from Bob by asking Alice to prove her identity by decrypting a chosen c & sharing result w/ Eve.

↑ chosen by Eve.

Bad

(Implement by Discord by Kassily) ← nice

Now we have another way for Alice to prove

Good

she owns an RSA key.

$D =$ "I am Alice"

Compute $D^{sig} = D^d$

Everyone else can compute $(D^{sig})^e =$ "I am Alice"

Numerical Example

Sam $p = 3$ $q = 5$. $N = 15$

$e = 3$ $e^{-1} \bmod (2)(4) = \underline{\underline{3 = d}}$

Public $K^{pub} = (15, 3)$

Signing $D = 7$

$D^{sig} = 7^3 = 13 \bmod 15$

Verifying

If Victor wants to check $(7, 13)$ is a valid pair, Victor computes

$(13)^e = 13^3 \equiv 7 \bmod 15$ ok cool!

Hard problem @ center of both is compute e'th root mod pq.

Digital Signatures using Discrete Logs

Slightly less straightforward than RSA.

Elgamal (1985) $\rightsquigarrow$ DSA (digital signature algorithm) (1991)

Rmk: The DLP for $\mathbb{F}_p^*$ is secure if $p \approx 2^{2000}$
"Index-Calculus Sieves present solution for smaller primes"

Elgamal Digital Signatures

① Key Creation

- Samantha (or trusted 3rd party) chooses a large prime p & a primitive root $g \in \mathbb{F}_p^*$
- Samantha choose a secret exponent a & computes

$\rightarrow A = g^a \bmod p$. ...

- Sam publishes $(A, g, p) = K^{Pub}$

② Signing

- Document D s.t. $1 < D < p$.
- Secret random number k s.t. $1 < k < p$ & $\gcd(k, p-1) = 1$
- Sam computes

$$S_1 \equiv g^k \bmod p \qquad S_2 \equiv (D - aS_1)k^{-1} \bmod p-1$$

- Sam signes $D^{sig} = (S_1, S_2)$

③ Verification

* Victor computes * $A^{s_1}\cdot S_1^{s_2}$ * g^{D} $\Big\}$ verifies these are congruent modulo p.

Proof of Correctness

Assume Sam signed D.

$$A^{s_1}\cdot S_1^{s_2} \equiv g^{as_1}\cdot g^{kS_2}$$

$$\equiv g^{as_1+kS_2}$$

$$\equiv g^{as_1+\underline{k}[(D-as_1)\underline{k^{-1}}]}$$

$kk^{-1}\equiv 1$ mod $p-1$

$$\equiv g^{as_1+D-as_1}$$

(FLT)

$$\equiv g^{D} \bmod p$$

$\varnothing$

Attacks

① Suppose Eve can solve the DLP.
The she can solve $g^x\equiv A \bmod p$ for $x=a$ & freely sign as Sam.

② This may not be the only way to forge Sam's signature.

Given $(A, g, P) = K^{pub}$

Eve wants to sign D.

Eve needs to provide som $D^{sig} = (x, y)$ so that Victors verification returns TRUE.

Checks if $A^x x^y \equiv g^D \mod p$ holds.

Problem becomes find x, y satisfying this

Apply $\log_g(\cdot)$

$$x \cdot \log_g A + y \cdot \log_g x \equiv D \mod p-1$$

Solve for $y = \dfrac{D - x\log_g A}{\log_g x}$

don't know w/o DLP.

Best known attack is DLP.

Remark $A^x \cdot x^y \equiv g^D \mod p$

Strange b/c x is an exponent (mod $p-1$) and also a base (mod p).

Subtleties

① One can easily produce x, y, D solving this problem

i.e. s.t. $A^x x^y \equiv g^D \mod p$.

Victor will verify.

D depends on x & y, so I can sign random documents.

HW

② If Sam ever uses the same k twice, eve can determine a & forge documents. (eve can easily tell when this happens too).

HW.

Sony hack

Remark

The pair $(S_1, S_2) \sim 2^{4000}$ bits is a pretty big. This can be improved.

Digital Signature Algorithm (DSA)

① Generating the Key

- Sam chooses p, q primes s.t. $p \equiv 1 \mod q$

  Rmk $p \approx 2^{2000}$, $q \approx 2^{200}$. Find this w/ slight modification to findPrime.

- Sam choose $g \in \mathbb{F}_p^*$ w/ $|g| = q$.

  Rmk $g_1 \in \mathbb{F}_p^*$ prim root, $g = g_1^{p-1/q}$ has order q

- Sam chooses secret exponent a.
- Sam computes $A = g^a \mod p$
- $K^{pub} = (A, p, q, g)$

② Signing

D document $1 < D < q$

- Sam choose random $k \in \mathbb{F}_q^*$ & compute

$S_1 = (g^k \bmod p) \bmod q$ $S_2 = (D + aS_1)k^{-1} \bmod q$

+ $D^{sig} = (S_1, S_2)$

③ Verification

Victor computes

$V_1 = DS_2^{-1} \bmod q$ $V_2 = S_1 S_2^{-1} \bmod q$

Victor Confirms

$$(g^{V_1} A^{V_2} \bmod p) \bmod q \equiv S_1 \bmod q$$

If it holds $\Longrightarrow$ TRUE.

Proof of Correctness

$$g^{V_1} A^{V_2} \equiv g^{DS_2^{-1}} g^{aS_1S_2^{-1}}$$

$$\equiv g^{(D+aS_1)S_2^{-1}}$$

$$\equiv g^{\underline{(D+aS_1)(D+aS_1)^{-1}}\cdot k}$$

$$\equiv g^k \bmod p$$

} reduce mod q

S_1

Note $|g| = q$

$\Rightarrow g^q = 1$

$ab \equiv 1 \bmod q$

$g^{ab} = g$

$\shortparallel \quad \shortparallel$

$g^{lq+1} = g^1$