MATH 314 Spring 2018 - Class Notes

11/28/2018

Scribe: Nicholas Visalli

Summary: Learn about digital analogue of signatures

Notes:

- Digital Signatures - Produce a number that somehow only the sender of a message can product
- Two Steps:

1. Signature Step
2. Verification Step

Both of these steps are used in a single algorithm - RSA Backwards

Setup:

- Alice produces an RSA public key
- She finds two primes p and q, n = pq
- She picks an e $\gcd(e, \Phi(n)) = 1, \Phi(n) = (p-1)(q-1)$
- She computes $d = e^{-1}(mod\Phi(n))$
- Public key: (n, e) ... d is secret

Now Alice wants to send a message m and prove that she is one sending it

Signing Step:

- She computes $S = m^d(modn)$
- She sends the pair (m, s) to Bob

Bob wants to verify Alice's signature

Bob obtains Alice's public key

Verification Step:

- Compute $S^e(modn)$

- Check to see if this is congruent to $mmodulon$
- If it is, he accepts the signature
- If not, he rejects it

Why does this work? $S^e \equiv (m^d)^e = m^{de} \equiv m(modn)$ And $de \equiv mod\Phi(n)$
Why can't Eve forge Alice's signature?

- Alice computes $S \equiv m^d(modn)$
- Eve would need to know d (But this requires factoring n, which is hard)
- This produces a number S from a message m in a way that only Alice can produce (knowing her secret key)

Mathematically tie a message to a person

- Could Eve come up with a different message signature?
- What if m can be big (bigger than n?)
- $m + n \equiv m(modn)$

So this message would have the same signature
$(m + n)^d \equiv m^d(modn) \equiv S(modn)$
S is connected to Alice but not to a unique m

We want to both hash and sign a message in order to tie the result to both the message and the person

Should we hash the signature or sign the hash?

Answer: Compute $S(h(m))$

In general, we don't sign messages directly. We sign the hash of the message (using a cryptographically secure hash)

Can we use the discrete log problem as a trapdoor function to sign messages?

One Way: DSA (Digital Signature Algorithm) uses the discrete log problem rather than factoring as a trapdoor

Setup:

- Alice picks a medium sized prime q (50 digits)

- Find a large prime p where q divides p-1. q, p-1

Example: $q = 13, p = 53 = 52 + 1 = 4(13) + 1$

Also need a primitive root $g(modp)$

- Compute $\alpha = (p-1) \div (q)(modp)$
- Note: $\alpha^q \equiv (g((p-1) \div (q))^q \equiv g^{p-1} \equiv 1(modp)$
- $\alpha^i \equiv \alpha^j(modp) <=> i \equiv j(modq)$

The point of this is that a lot of arithmetic can happen $(modq)$ makes things faster

- Alice picks a secret exponent a $(1 < a < q-1)$
- She computes $\beta \equiv \alpha^a(modp)$
- Her public key is (p, q, α, β)

Alice wants to sign a message m
Signature Step:

- She picks a new random number k $(1 < k < a-1)$
- She computes $r \equiv (\alpha^k(modp))(modq)$
- $S = k^{-1}(m + ar)(modq)$
- She sends (m, r, s)

Bob wants to verify their signature

Verification Step:

- He computes $U1 \equiv S^{-1}m(modq)$, $U2 \equiv S^{-1}r(modq)$
- Then he computes $v \equiv (\alpha^{u1}\beta^{u2}(modp)(modq))$
- If $v \equiv r(modq)$, he accepts the signature
- If not, he rejects it

Why does this work?

- Recall $S \equiv K^{-1}(m + ar)(modq)$
- $KS \equiv m + ar(modq)$
- $K \equiv S^{-1}m + S^{-1}(ar)(modq)$

- $= U1 + \alpha U2 (mod q)$
- $\alpha^k \equiv \alpha^{u1+u2} \equiv \alpha^{u1}(\alpha^q)^{u2} \equiv \alpha^u 1 \beta^u 2 (mod p)$

Most of the arithmetic happens $(mod q)$

This is fast

If Eve wants to break this, she would need to solve the discrete log problem (mod p) makes this secure.