MATH 314 Spring 2020 - Class Notes

5/4/2019

Scribe: Matthew Clark

Summary: Discuss two different mathematical tools used for verifying the identity of a sender when receiving encrypted data, AKA digital signature. In the discussion we review the process on how to mathematically tie a signature to an individual

Notes: Include detailed notes from the lecture or class activities. Format your notes nicely using latex such as

- If Bob receives a message from Alice how can he tell if it is really from or Alice or just Eve pretending to be Alice?
- Physical World: If you receive a letter from Alice, her signature at the bottom guarantees it is from her
- For the digital world: replace this idea using math instead
- Two things our signature is connected to:
  1. It can only be produced by 1 person
  2. Physically connected to the message

Today: How to mathematically the signature tie a signature to an individual known as digital signatu

Digital signatures

- Produce a number that could only have been produced by the person who owns the signature
- Also need a way to verify that the signature is valid
- To do this we need a public/private key
- Signature function use the $S_k(m)$ a private key to generate a signature for the message m
- Verification function $V_k(m, s)$which uses the public key to test whether s is a valid signature for the message m
- RSA digital signature - set up is exactly the same as regular RSA
- Alice finds n =pq and e where gcd(e, phi(n)) = 1

- Her public key is (n,e) —— secretly Alice computes $d = e^{-1}(mod L(n))$
- Only Alice can compute d because finding phi(n) = (p-1)(q-1) requires knowing p and q
- Now Alice wants to send an encrypted message m to Bob along with a signature to prove it comes from Alice
- Alice "signs" the message me using the signature function $S = S_d(m) = m^d(mod n)$
- Alice sends the message (m,s) to Bob
- Bob gets (m,s) but really isn't sure if it comes from Alice, he needs a way of checking
- To verify s Bob uses Alice's public key —— He computes $S^e(mod n)$ —— if it equals m(mod n) Bob accepts the signature as valid, otherwise he rejects it as forgery
- Why should we $s^e = m(mod n)$ if the signature is valid? —— If Alice produced s using the private key then $s = m^d$
- so $s^e = (m^d)^e = m^{ed} = m(mod n)$ —— same as decryption!
- Why can't Eve forge Alice's signature on another message m?
- If Eve just picks any number $s^{'}$ and sends $(m^{'}, s^{'})$ to Bob then $s^d /= m(mod n)$
- To make it valid she would need to find a number $s^{'}$ where $s^{'e} = m^{'}(mod n)$
- The only $s^{'}$ that works is $s^{'} = m^{'d}(mod n)$ —— the only way to compute that is to know d and figuring out d is hard

Digital Signature Algoirthm (DSA)

- uses the discrete log problem as a one way function
- similar to el gamal
- Set up for DSA:
- Large prime p and medium prime q. Try to do the most arithmetic (mod q) fast, get most of the security of working mod a large p
- Need q to dive p-1 —— Ex: p = 101 q = 5 —— s divides p-1 = 100
- g - primitve root(mod p)
- alp = $g^{(p-1)/a}$(mod p) integer since q divides p-1
- Alice picks a secret number a $2 <= a < q - 1$

- $B = alp^a$(mod p)
- Alice public key is all four numbers (p,q,alp,B)
- Alice wants to send a message M with a DSA signature that proves it her message
- 1st Alice a ephemeral key k: $2 < k < q - 1$
- $r = (s^k modp)(modq)$ —— $s = k^{-1}(m + ar)(modq)$ —— (r,s) is Alices signature
- She sends (m(r,s)) to bob
- bob wants to verify this signature —— $U_1 = s^{-1}(m)(modq)$ —— $U_2 = s^{-1}(r)(modp)$
- $V = (alp^{u_1} * b^{u_2}(modp))(modq)$
- if V = r he accepts the signature as valid, otherwise reject it
- Why should V = r(mod q)???
- $s = k^{-1}$(m +ar)(mod q)
- $k = s^{-1}$(m + ar)(mod q)