DTTF/NB479: Dszquphsbqiz Day 29

- Announcements:
- Questions?
- This week:
  - Digital signatures, DSA
  - Flipping coins over the phone

Why are digital signatures important?

- Compare with paper signatures
- Danger: Eve would like to use your signature on other documents!
- Solution: sig = f(m, user)
  - Let m be the message (document)
- Algorithms we'll study:
  - RSA
  - ElGamal
  - DSA (Digital Signature Algorithm)

RSA Signatures

- Alice chooses:
  - p,q, n=pq,
  - e: gcd(e, (p-1)(q-1))=1,
  - d: ed ≡ 1(mod ((p-1)(q-1)) [d is the "pen" Alice uses]
- Publishes n, e ["glasses" Bob uses to see the writing]
- Alice's signature uses the *decryption exponent*:
  - $y \equiv m^d \pmod n$. Delivers (m, y)
- Bob's verification:
  - Does $m \equiv y^e \pmod n$?
- Show the verification works. (Q1)
- Note that given only the signature y, and public info e and n, Bob can compute the message, m.

RSA Signatures

- Alice chooses:
  - p,q, n=pq,
  - e: gcd(n, (p-1)(q-1))=1,
  - d: ed ≡ 1(mod ((p-1)(q-1))
- Publishes n, e
- Alice's signature:
  - $y \equiv m^d \pmod{n}$. Delivers (m, y)
- Bob's verification:
  - Does $m \equiv y^e \pmod{n}$?

- Eve's schemes:
  - Can she use Alice's signature on a different document, m_1?
  - Can she compute a new y_1, so that $m_1 = y_1^e$?
  - Can she choose a new y_1 first, then compute $m_1 = y_1^e$?

Blind Signature

- Alice chooses:
  - p,q, n=pq,
  - e: gcd(n, (p-1)(q-1))=1,
  - d: ed ≡ 1(mod ((p-1)(q-1))
- Publishes n, e
- Bob wants m signed
- Bob chooses:
  - k: random, gcd(k, n)=1
- Bob sends: $t \equiv k^e m \pmod{n}$
- Alice's signature:
  - $s \equiv t^d \pmod{n}$.
- Bob's verification:
  - Computes sk^{-1}

- Bob wants Alice to sign a document as a method of time-stamping it, but doesn't want to release the contents yet.*

- Verification:
  - Find sk^{-1} in terms of m
  - What is the significance of this?

- Why can't Alice read m?
- What's the danger to Alice of a blind signature?

* He can publish her signature, which can be verified later, or he can submit it to an authority to obtain an actual timestamp: http://en.wikipedia.org/wiki/Trusted_timestamping

ElGamal Signatures don’t reveal the message during verification

- Many different valid signatures for a given message
- Alice chooses:
  - p,primitive root α, $\beta \equiv \alpha^a \pmod{p}$
  - Publishes (p, α, β), keeps a secret
- Alice’s signature:
  - Chooses k: random, gcd(k, p-1)=1
  - Sends (m, (r,s)), where:
    - $r \equiv \alpha^k \pmod{p}$
    - $s \equiv k^{-1}(m - ar) \pmod{p-1}$
- Bob’s verification:
  - Does $\beta^r r^s \equiv \alpha^m \pmod{p}$?

ElGamal Signatures

- Many different valid signatures for a given message
- Alice chooses:
  - p,primitive root α, secret **a**, and $\beta \equiv \alpha^a \pmod p$
  - Publishes (p, α, β), keeps **a** secret
- Alice's signature:
  - Chooses k: random, gcd(k, p-1)=1
  - Sends m, (r,s), where:
    - $r \equiv \alpha^k \pmod p$
    - $s \equiv k^{-1}(m - ar) \pmod{p-1}$
- Bob's verification:
  - Does $\beta^r r^s \equiv \alpha^m \pmod p$?

- Notice that one can't compute m from (r,s).
- Show the verification works.
- Why can't Eve apply the signature to another message?
- If Eve learns a, she can forge the signature
- Note: Alice needs to randomize k each time, else Eve can recognize this, and can compute k and a relatively quickly.