Introduction to Modern Cryptography

Lecture 5

- Number Theory:
    1. Quadratic residues.
    2. The discrete log problem.

- Intro to Public Key Cryptography
- Diffie & Hellman Key Exchange

Course Summary - Math Part (first 4 lectures)

- Euclid gcd ; extended gcd.
- The ring Z_m.
- Finite groups: Lagrange theorem (if G is finite and H is a sub-group then $|H|$ divides $|G|$)
- Finite fields arythmetic - $GF(p^k)$.
- Primitive elements in finite fields (generators of the multiplicative group with p^k-1 elements)
- The birthday paradox.

Course Summary - Crypto Part (first 4 lectures)

- Introduction
- Stream & Block Ciphers
- Block Ciphers Modes (ECB,CBC,OFB)
- Advanced Encryption Standard (AES)
- Message Authentication Codes (based on CBC and on cryptographic hashing)

The Birthday Paradox: Wrap Up

- Let R be a finite set of size r.
- Pick k elements of R uniformly and independently.
- What is the probability of getting at least one collision ?

The Birthday Paradox (cont.)

- Consider the event E_k: No Collision after k elements.

  $\text{Prob}(E_k) = 1(1- 1/r)(1- 2/r)\ldots (1- (k-1)/r)$

  $< \exp(-1/r)\exp(-2/r)\ldots\exp(-(k-1)/r)$

  $= \exp(-(1+2+\ldots+(k-1))/r)$

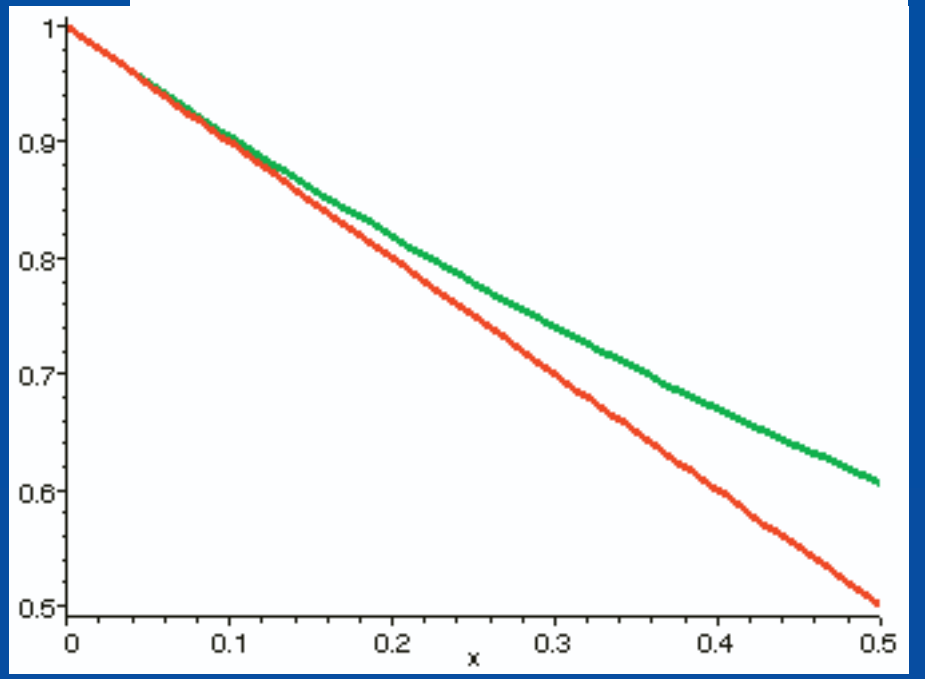
  $= \exp(-(k(k-1))/2r)$

  $\sim \exp(-k^2/2r)$

For $k=r^{1/2}$, $\text{Prob}(E_k)<0.607$,
thus $\text{Prob}(\text{Collision}_k)>0.393$

For $k=1.2r^{1/2}$, $\text{Prob}(E_k)<0.487$,
thus $\text{Prob}(\text{Collision}_k)>0.513$

plot({exp(-x),1-x},x=0..0.5);



Application to Cryptographic Hashing

Let H:D --> R, R of size r.

Suppose we can get k random images under H.

If k^2 is larger than r then the probability of a collision, $1-\exp(-k^2/2r)$, is large.

Thus a necessary condition for avoiding collisions is that r is so large that it is infeasible to generate r^2 hash values.

This leads to requiring that message digests be at least 160 bits long ($2^{160/2} = 2^{80}$ is large enough).

Back to Number Theory

Fermat "Little" Theorem

- Let if G be a finite group with m elements. Let a be an element of G. Then $a^m = 1$ (the unit element of G).

Example

$G=Z_p^*$, the multiplicative group of Z_p. The polynomial $x^{p-1}-1$ has $p-1$ roots, so $x^{p-1}-1 = \Pi_{a \neq 0}(x-a)$.

```
> factor (x^6-1);
   (x-1)(x+1)(x^2+x+1)(x^2-x+1)

> factor (x^6-1) mod 7;
    (x+6)(x+1)(x+4)(x+2)(x+5)(x+3)
```

Quadratic Residues

- Definition: An element x is a *quadratic residue* modulo n if there exists y such that $y^2 \equiv x \bmod n$
- Claim: if p is prime there are exactly $(p-1)/2$ quadratic residues in Z_p^*
- Claim: if p is prime, and g is a generator of the multiplicative group, the quadratic residues are all the even powers of g

  $g^0, g^2, \ldots, g^{2i}, \ldots, g^{p-3}$

Quadratic Residues in Z_p (cont.)

- The quadratic residues (QR) form a subgroup of Z_p^*.
- $x^{(p-1)} - 1 = (x^{(p-1)/2} - 1)(x^{(p-1)/2} + 1)$.
- Thus $x^{(p-1)/2} - 1$ has $(p-1)/2$ roots in Z_p.

Quadratic Residues in Z_p (cont.)

Claim: an element x in Z_p is a quadratic residue if and only if $x^{(p-1)/2} \equiv 1 \bmod p$

Proof Sketch:

- Suppose $x=y^2$ (x is a QR), then $x^{(p-1)/2} - 1 = 0$.
- Suppose $x^{(p-1)/2}=1$. Let $x=g^i$ where g is primitive element. Then $g^{i\,(p-1)/2} = 1$. Since g has order $p-1$, $p-1$ must divide $i\,(p-1)/2$, implying i even, x a QR.

Testing Quadratic Residues

- Efficient $O(\log^3 p)$ algorithm in Z_p (p prime)
- Applies the repeated squaring idea.

- For composite m (esp. m=pq), no efficient algorithm for testing quadratic residues is known. Problem believed to be computationally hard (but not NPC).

Discrete Log (DL)

- Let G be a group and g an element in G.
- Let $y=g^x$ and x the minimal non negative integer satisfying the equation.
- x is called the *discrete log* of y to base g.
- Example: $y=g^x$ mod p in the multiplicative group of Z_p

Discrete Log in Z_p
A candidate for One Way Function

- Let $y=g^x \bmod p$ in the multiplicative group of Z_p
- Exponentiation takes $O(\log^3 p)$ steps
- Standard discrete log is believed to be computationally hard.
- $x \longrightarrow g^x$ is easy (efficiently computable).
- $g^x \longrightarrow x$ believed hard (computionally infeasible).

- $x \longrightarrow g^x$ is a one way function.
- This is a computation based notion.

Public-Key Cryptography

The New Era (1976-present)

Classical, Symmetric Ciphers

- Alice and Bob share the same secret key $K_{A,B}$.

- $K_{A,B}$ must be secretly generated and exchanged prior to using the unsecure channel.

Diffie and Hellman (76)
"New Directions in Cryptography"

Split the Bob's secret key K to two parts:

- K_E , to be used for encrypting messages to Bob.
- K_D , to be used for decrypting messages by Bob.

K_E can be made public
(public key cryptography,
assymetric cryptography)

"New Directions in Cryptography"

- The Diffie-Hellman paper (IEEE IT, vol. 22, no. 6, Nov. 1976) generated lots of interest in crypto research in academia and private industry.
- Diffie & Hellman came up with the revolutionary idea of public key cryptography, but did not have a proposed implementation (these came up 2 years later with Merkle-Hellman and Rivest-Shamir-Adelman).
- In their 76 paper, Diffie & Hellman did invent a method for key exchange over insecure communication lines, a method that is still in use today.

Public Exchange of Keys

- Goal: Two parties (Alice and Bob) who do not share any secret information, perform a protocol and derive the same shared key.
- Eve who is listening in cannot obtain the new shared key if she has limited computational resources.

Diffie-Hellman Key Exchange

- Public parameters: A prime p, and an element g (possibly a generator of the multiplicative group Z_p^*)
- Alice chooses a at random from the interval $[1..p-2]$ and sends $g^a \bmod p$ to Bob.
- Bob chooses b at random from the interval $[1..p-2]$ and sends $g^b \bmod p$ to Alice.
- Alice and Bob compute the shared key $g^{ab} \bmod p$:

  Bob holds b, computes $(g^a)^b = g^{ab}$.

  Alice holds a, computes $(g^b)^a = g^{ab}$.

DH Security

- DH is at most as strong as DL in Z_p.
- Formal equivalence unknown, though some partial results known.
- Despite 25 years effort, still considered secure todate.

- Computation time is $O(\log^3 p)$.

Properties of Key Exchange

- Necessary security requirement: the shared secret key is a one way function of the public and transmitted information.
- Necessary "constructive" requirement: an appropriate combination of public and private pieces of information forms the shared secret key efficiently.
- DH Key exchange by itself is effective only against a passive adversary. Man-in-the-middle attack is lethal.

Security Requirements

- Is the one-way relationship between public information and shared private key sufficient?
- A one-way function may leak some bits of its arguments.
- Example: $g^x \bmod p$
- Shared key may be compromised
- Example: $g^{x+y} \bmod p$

Security Requirements (cont.)

- The full requirement is: given all the communication recorded throughout the protocol, computing *any* bit of the shared key is hard
- Note that the “any bit” requirement is especially important

Other DH Systems

- The DH idea can be used with any group structure
- Limitation: groups in which the discrete log can be easily computed are not useful
- Example: additive group of Z_p
- Currently useful DH systems: the multiplicative group of Z_p and elliptic curve systems

Key Exchange in Systems

- VPN usually has two phases
  - Handshake protocol: key exchange between parties sets symmetric keys
  - Traffic protocol: communication is encrypted and authenticated by symmetric keys
- Automatic distribution of keys- flexibility and scalability
- Periodic refreshing of keys- reduced material for attacks, recovery from leaks