Public-Key Cryptosystems

CS 6750 Lecture 4

October 1, 2009

Riccardo Pucella

Problems with Shared Keys

- All cryptosystems we have looked at until now have required a shared key between senders and receivers
- Problems:
  - How do you establish the keys and distribute them?
  - In a network of N people, need N^2-N keys total
  - Any new person joining requires creating and distributing N new keys.
- Solutions:
  - Figure out how to distribute keys easily
  - Find an altogether different approach

Public-Key Cryptography

- Diffie and Hellman (1976) proposed a scheme where keys need not be shared
  - Idea: provide every agent with two different keys
  - One key is used to encrypt
  - One key is used to decrypt
  - The key to encrypt is made public
  - The key to decrypt is kept private (secret)
- Anyone can send an encrypted message to Alice by using her public encryption key
- Only Alice can read the encrypted message because she has the private decryption key

One-Way Trapdoor Functions

- For this to work, need a way to find encryption and decryption keys such that knowing the encryption key does not let you derive the decryption key
- Diffie and Hellman's idea: one-way trapdoor functions
  - One-way: a function whose inverse is hard to compute
  - Trapdoor: but if you have a specific hint, you can invert the function easily
- To encrypt, apply the one-way function
- To decrypt, use the hint to invert the function
- Challenge: are there any one-way trapdoor functions?

Candidates

- Two most likely one-way trap door function candidates:
  - Factorization → RSA cryptosystem
  - Discrete logarithms → ElGamal cryptosystem
- No one has ever proved that these are one-way trapdoor functions
  - It's proving that they are one-way that's a problem
- In fact, no one knows for sure that there exists a one-way function
- All known candidates involve number theory or algebra

Number Theory on a Slide

- Recall: $ax \equiv 1 \bmod n$ has a solution for x iff $\gcd(a,n)=1$
- $\varphi(n)$ = # of integers $k<n$ such that $\gcd(n,k)=1$

- Define $Z_n^* = \{a : \gcd(a,n)=1\}$
  - For prime p, $Z_p^*=\{1,\ldots,p-1\} = Z_{p-1}$

- If we define $ab = ab \pmod n$, then Z_n^* is an Abelian group under multiplication
  - i.e., behaves like integers under addition

- Theorems:
  - If $b \in Z_n^*$ then $b^{\varphi(n)} \equiv 1 \bmod n$
  - If p is prime and $b \in Z_p^*$ then $b^p \equiv b \bmod p$

RSA Cryptosystem

- Rivest, Shamir and Adleman (1978)
  - Some classified independent work in the UK in 1973

- Take $n = pq$ (where p and q are primes)
- $P = C = Z_n$
- $K = \{(n,p,q,a,b) : ab \equiv 1 \bmod \varphi(n)\}$
- For $k=(n,p,q,a,b)$
  - $e_k(x) = x^b \pmod n$ - need only n,b
  - $d_k(y) = y^a \pmod n$

- Choose p,q large, compute $n=pq$. $\varphi(n)=(p-1)(q-1)$
  Choose b with $\gcd(b,\varphi(n))=1$
  Let $a=b^{-1} \pmod{\varphi(n)}$, publish n,b and keep p,q,a private

Sanity Check

- Need to check that encryption and decryption are inverses
- Let $x \in Z_n^*$ (slightly different argument if $x \in Z_n - Z_n^*$)
  - Exercise: derive that $d_k(e_k(x)) = x$
- Hint: Since $ab \equiv 1 \bmod \varphi(n)$, then $ab = t\varphi(n)+1$ for some $t \geq 1$

Security of RSA

- Security of RSA based on the belief that e_k is a one-way function
  - Strong evidence, but we don't know for sure
  - It is a trapdoor function. What's the hint? The factorization n=pq. With a,n,p,q, can recover b by taking $b = a^{-1} \pmod{\varphi(n)}$

- Need n to be hard to factor into p,q -- p and q in practice need to be large enough (512 bits and more)

- How do you find primes of this size?
  - Best: generate numbers randomly, and test primality
  - Chance of finding a prime ~ 1/355
  - Primality testing can be done fast (Stinson §5.4)

Attacks Against RSA

- Factoring attacks
  - HUGE literature -- see Stinson §5.6

- Compute φ(n) directly from n
  - No easier than factoring
    - If you have n and φ(n), it is almost trivial to get factorization by solving:

$$n = pq \quad \rightarrow \quad q = n/p$$

$$\varphi(n) = (p-1)(q-1) \quad \rightarrow \quad \varphi(n) = (p-1)(n/p-1)$$

- Find a directly?
  - Can also show that given a and n you can find the factorization p,q

Discrete Logarithms

- Let G be any multiplicative group (e.g., Z_n^*)
  - The order of an element $\alpha \in G$ is the smallest n with $\alpha^n=1$ in G
  - Given $\alpha \in G$ of order n, $\langle\alpha\rangle=\{\alpha^0,\alpha^1,\ldots,\alpha^{n-1}\}$
  - $\langle\alpha\rangle$ is a subgroup of G
  - α is a primitive element of G if $\langle\alpha\rangle = G$

- Given G a multiplicative group, $\alpha \in G$ of order n, $\beta \in \langle\alpha\rangle$: the discrete logarithm of β is the unique integer $d<n$ with $\alpha^d = \beta$ in G

Discrete Logs in Z_p^*

- Why are discrete logs interesting?
  Computing discrete logs is believed to be hard for some multiplicative groups

- Theorem:
  $Z_n^* = \langle\alpha\rangle$ for some $\alpha \in Z_n^*$

- The ElGamal cryptosystem is based on discrete logs in Z_p^* for some prime p
  - Believed to be hard for Z_p^* with $p > 300$ digits and $p-1$ with at least one large prime factor

ElGamal Cryptosystem

- Let p be a prime such that discrete logs in Z_p^* are believed hard to compute
- Let α be a primitive element of Z_p^*
- $P = Z_p^*$
- $C = Z_p^* \times Z_p^*$
- $K = \{(p,\alpha,d,\beta) : \beta = \alpha^d \pmod p\}$
- For $k=(p,\alpha,d,\beta)$
  - $e_k(x,k) = (\alpha^k \pmod p,\ x\beta^k \pmod p)$ - need only p,α,β
    for some $k \in Z_p^*$ chosen at random
  - $d_k(y_1,y_2) = y_2(y_1^d)^{-1} \pmod p$

- Chose α and d, compute $\beta=\alpha^d \pmod p$
  Publish p,α,β, keep d private

ElGamal Cryptosystem

- Let p be a prime such that discrete logs in Z_p^* are believed hard to compute
- Let α be a primitive element of Z_p^*
- $P = Z_p^*$
- $C = Z_p^* \times Z_p^*$
- $K = \{(p,\alpha,d,\beta) : \beta = \alpha^d \pmod{p}$
- For $k=(p,\alpha,d,\beta)$
  - $e_k(x,k) = (\alpha^k \pmod{p}, x\beta^k \pmod{p})$ - need only p,α,β
    for some k
  - $d_k(y_1,y_2) = y_2(y_1^d)^{-1}$
- Chose α and d, comp
  Publish p,α,β, keep d priva

Hide x with β^k

pass k along as α^k

Sanity Check

- Need to check that encryption and decryption are inverses

- Exercise: derive that $d_k(e_k(x,k)) = x$, for any k
  - E.g., $d_k(\alpha^k \pmod p, x\beta^k \pmod p) = x$

- Given p,α,β, the attacker "needs" to compute d such that $\alpha^d \equiv \beta \pmod p$
  - Stinson §6.2 and §6.3 present some of the best known algorithms to find discrete logs

Elliptic Curves

- The ElGamal cryptosystem can be implemented in any group where the discrete log problem is (believed to be) difficult
    - Historically, Z_p^* has been used
- Other groups have become popular
- Elliptic curves modulo a prime p>3:
    - Let $a,b \in Z_p$ such that $4a^3+27b^2 \neq 0$
    - A nonsingular elliptic curve modulo p is the set of all E of all $x,y \in Z_p$ such that $y^2 \equiv x^3+ax+b \mod p$ (plus a special point O -- the point at infinity)
    - E is a group by defining an operation + on points

Some Conclusions

- Public-key cryptography solves the key distribution problem by eliminating it
    - Public keys are published in some repository
    - Private keys are kept private
- Comes at a cost: public-key cryptography is much slower than shared-key cryptography (such as DES)
    - Not ideal for long messages
- Hybrid solution (PGP-style):
    - Alice wants to communicate with Bob
    - Alice creates a shared key, sends it to Bob via a public-key cryptosystem
    - Alice sends message to Bob via the shared key