Public key crypto (quick intro)
Provable cryptography

Slides from Bart Preneel and Phil Rogaway
Comp527 status

• Hack-a-Vote phase 2 complete
  – Scott will make everything public
  – See what you missed / what others found
• Phase 3 now assigned
  – Use cryptyc to model a better crypto protocol
  – Scott’s tutorial from Monday online later today
Public key primitives

• Diffie-Hellman
  – Hard problem: Discrete logarithms

• RSA
  – Hard problem: Factoring composite numbers

• Field: integers modulo a large prime number (numbers wrap around)
A public-key distribution protocol: Diffie-Hellman

- Before: Alice and Bob have never met and share no secrets; they know a public system parameter $\alpha$

  generate $x$

  compute $\alpha^x$

  $\alpha^x$

  $\alpha^y$

  compute $k=(\alpha^y)^x$

  compute $k=(\alpha^x)^y$

- After: Alice and Bob share a short term key $k$
  - Eve cannot compute $k$: in several mathematical structures it is hard to derive $x$ from $\alpha^x$ (this is known as the discrete logarithm problem)
Diffie-Hellman (continued)

\[ \begin{align*} 
&\text{generate } x \\
&\text{compute } \alpha^x \\
&\text{compute } k = (\alpha^y)^x \\
\end{align*} \]

\[ \begin{align*} 
&\text{generate } y \\
&\text{compute } \alpha^y \\
&\text{compute } k = (\alpha^x)^y \\
\end{align*} \]

- BUT: How does Alice know that she shares this secret key \( k \) with Bob?
- Answer: Alice has no idea at all about who the other person is! The same holds for Bob.
Station to Station protocol (STS)

- The problem can be fixed by adding digital signatures
- Many variations on this theme used in practice

\[
\begin{align*}
\text{choose } x &\quad \alpha^x \\
\alpha^y &\quad \text{choose } y \\
k = (\alpha^y)^x &\quad \text{Sig}_A(\alpha^x, \alpha^y) \\
\sqrt{\text{Sig}_B} &\quad \text{Sig}_B(\alpha^y, \alpha^x)
\end{align*}
\]
Footnote: if you can define multiplication...

- “Elliptic curve” crypto looks the same as Diffie-Hellman
- Instead of integers mod $N$
  - $y^2 = x^3 + Ax^2 + B \pmod{p}$
  - $A, B, p$ are “carefully chosen”
  - Integers $(x,y)$ on the curve form a group
  - Addition, multiplication, exponentiation can be defined

- Claim: DLog is harder for elliptic curves than modular integer arithmetic
  - Therefore we can use smaller numbers $\Rightarrow$ faster computation
RSA (‘78)

• Choose 2 “large” prime numbers \( p \) and \( q \)
• modulus \( n = p.q \)
• compute \( \lambda(n) = \text{lcm}(p-1,q-1) \)
• choose \( e \) relatively prime w.r.t. \( \lambda(n) \)
• compute \( d = e^{-1} \mod \lambda(n) \)

• public key = \((e,n)\)
• private key = \((d,p,q)\)

The security of RSA is based on the “fact” that it is easy to generate two large primes, but that it is hard to factor their product

- encryption: \( c = m^e \mod n \)
- decryption: \( m = c^d \mod n \)

try to factor 2419
What about quantum computers?

- exponential parallelism

\[ n \text{ coupled quantum bits} \]

\[ 2^n \text{ degrees of freedom!} \]

- Shor 1994: perfect for factoring
- But: can a quantum computer be built?
State of the art in coherent qubit control

- Stanford/IBM NMR, main players
  - Other NMR non-NMR
  - Grover search 280 2-bit gates
  - 99 Oxford
  - 98
  - 99, Oxford
  - 99, 01
  - 99, 00, 01 MIT
  - 98 MIT
  - 98 Cambridge
  - 00, 01
  - 98, LNAL

- Liquid crystals
  - 99

- Error detection
  - 01 NEC
  - 02 Sacley
  - 99 NEC
  - 95 NIST
  - 95 Caltech

- “Cooling” spins
  - 99, 01
  - 99, 00, 01 MIT
  - 98 MIT
  - 98 Cambridge
  - 00, 01
  - 98, LNAL

- Deutsch-Jozsa
  - 00 NIST

- Error correction
  - 01 LANL
  - 01 Frankfurt
  - 00 LANL
  - Shor 15 = 3x5

- 7-spin coherence
  - 99 Cambridge

* unpublished
Advantages of public-key cryptology

• Reduce protection of information to protection of authenticity of public keys

• Confidentiality without establishing secret keys
  – extremely useful in an open environment

• Data authentication without shared secret keys: digital signature
  – sender and receiver have different capability
  – third party can resolve dispute between sender and receiver
Disadvantages of public-key cryptology

• Calculations in software or hardware two to three orders of magnitude slower than symmetric algorithms
• Longer keys: 1024 bits rather than 56…128 bits
• What if factoring is easy?
Practical Cryptography:
Provable Security as a Tool for Protocol Design

Phillip Rogaway
UC Davis & Chiang Mai Univ
rogaway@cs.ucdavis.edu
http://www.cs.ucdavis.edu/~rogaway

Summer School on Foundations of Internet Security
17-19 June 2002
Duszniki Zdroj, Poland
(three two-hour lectures)

Slides modified and tweaked by Dan Wallach, with permission
0. Opening comments
1. What is "provably security"?
2. Blocks ciphers
   2.1 Syntax
   2.2 Notions of security (prp, prf, kr)
3. Symmetric Encryption
   3.1 Syntax
   3.2 Notions of security (sem, ind, ind$, all under CPA)
4. Relating the notions (ind$, ind, 01)
5. Sample block-cipher-using encryption schemes
6. Security of modes
   6.1 CTR-rand
   6.2 CBC-rand
7. MACs and authenticated encryption
   7.1 Notion of authenticated encryption
   7.2 Notion of MACs
   7.3 Ways to MAC (CBC, XCBC, CW (w/ poly-based universal hash, UMAC)
   7.4 Ways to achieve auth enc (generic composition, IAPM/OCB)
Concluding comments
Recognize Problem
  └── Protocol
      └── Bug
           └── New Protocol
               └── Bug
                   └── New Protocol
                       └── Publish
                           └── Implement
                                └── Ship
                                    └── Bug

“Classical Approach”
“Provable-Security Approach” begins with [GM82]
If primitive $\pi$ is secure then protocol $\Pi$ is secure.
If $\not\exists$ a good adv for attacking $\pi$ then $\not\exists$ no good adv for attacking $\Pi$.
If $\not\exists$ a good adv for attacking $\Pi$ then $\not\exists$ a good adv for attacking $\pi$. 
**Block-Cipher Syntax**

\[ E: K \times \{0,1\}^n \rightarrow \{0,1\}^n \]

where each \( E_K(\cdot) = E(K, \cdot) \) is a permutation

Eg: \( E_K(X) = X \)
\( E_K(X) = AES_{128K}(X) \)
Notions of Block-Cipher Security

Key-recover (kr) under chosen-plaintext attack (CPA)

\[
\text{Adv}^{\text{kr}}_E (A) = \Pr [ K \overset{\$}{\leftarrow} K : A^{E(K, \cdot)} = K]
\]

\[
\text{Adv}^{\text{kr}} (t, q) = \max \{ \text{Adv}^{\text{kr}}_E (A) \}
\]

Adv runs in time \( \leq t \)
Asks \( \leq q \) queries
PRP-sense of a block cipher being good
\[ \text{Adv}^{\text{prp}}_{E}(A) = \Pr \left[ K \leftarrow S: A^{E(K, \cdot)} = 1 \right] - \Pr \left[ \pi \leftarrow \text{Perm}(n): A^{\pi(\cdot)} = 1 \right] \]

\[ \text{Adv}^{\text{prp}}_{E}(t,q) = \max \{ \text{Adv}^{\text{prp}}_{E}(A) \} \]

Attacker A responds:
0: it’s a permutation
1: it’s the cipher

Runs in time \leq t
Asks \leq q queries
Breaking $E_K(X)=X$

A: Ask $0^n$, receiving $Y$
   if $Y=0^n$ return 1  (cipher returns the identity)
   else return 0

$\text{Adv}^{\text{prp}}_E(A) = 1 - 2^{-n}$  (permutation might also)

$\text{Adv}^{\text{prp}}_{\text{AES}}(t,q) \leq t / 2^{128}$  Strong assumption

$\text{Adv}^{\text{prp}}_{\text{AES}}(t,q) \leq 2^{-40}$ if $t<2^{80}$, $q<2^{40}$  Weaker assumption
\[
\text{Adv}^\text{prf}_E(A) = \Pr \left[ K \xleftarrow{\$} K: A^{E(K, \cdot)} = 1 \right] - \\
\Pr \left[ \rho \xleftarrow{\$} \text{Rand}(n): A^{\rho(\cdot)} = 1 \right]
\]

\[
\text{Adv}^\text{prf}_E(A) = 2\Pr \left[ b \xleftarrow{\$} \{0,1\} ; \\
\text{if } b=1 \text{ then } K \xleftarrow{\$} K, f=E_K \text{ else } f \xleftarrow{\$} \text{Rand}(n): A^{f(\cdot)} = b \right] - 1
\]
“Switching Lemma”  If A asks $\sigma$ queries

$$|\text{Adv}_{E}^{\text{prp}}(A) - \text{Adv}_{E}^{\text{prf}}(A)| \leq \sigma^2 / 2^{n+1}$$

$$\Pr[A^{\pi(\cdot)} = 1] - \Pr[A^{\rho(\cdot)} = 1] \leq \sigma^2 / n+1$$
**Def.** A (sym, prob) enc scheme is a 3-tuple
\( \Pi = (K, E, D) \)

Finite set

\( E : K \times M \rightarrow \{0,1\}^* \) is a prob. function

\( D : K \times \{0,1\}^* \rightarrow M \cup \{\ast\} \) (det funct)

If \( M \in M \) and \( |M'| = |M| \), then \( M' \in M \)

\( M \in M, K \in K, C \leftarrow E_K(M) \Rightarrow D_K(C) = M \)

\( |C| = \text{clen}(|M|) \)
support(\( M \)) only has strings of one length

\[ \Pi = (K, E, D) \]

\[
\text{Adv}_{\Pi}^{\text{sem}}(A) = \Pr \left[ K \leftarrow K; \ (f, M) \leftarrow A^{E(K, \cdot)}(\cdot); M \leftarrow M; C \leftarrow E_K(M): \right. \\
A^{E(K, \cdot)}(C, f') = f(M) \right] - \\
\Pr \left[ K \leftarrow K; \ (f, M) \leftarrow A^{E(K, \cdot)}(\cdot); M, M' \leftarrow M; C \leftarrow E_K(M'); \right. \\
A^{E(K, \cdot)}(C, f) = f(M) \right]
\]