A Introduction to Modern Cryptography

smb@research.att.com

http://www.research.att.com/~smb

973-360-8656

AT&T Labs Research

Florham Park, NJ 07932

What is Cryptography?

- The art of “Secret Writing”
- A set of mathematical functions
- The basis for lots of cool tricks

Classic Ciphers

- Encrypted messages composed of letters
- In the telegraph era, produced letters; before that, some ciphers produced weird symbols
- Example: “Caesar cipher”

  $$A \rightarrow D, B \rightarrow E, \ldots$$

  More generally, replace each letter with the one n further down the alphabet, wrapping around if necessary.

The System versus the Key

- The general *system* here is “replace a letter by one further down in the alphabet”.
- The *key* is the amount to shift: 3 in this case.
- Assume that the enemy knows the system but not the key.

Cryptography Becomes Mathematical

- In the 1920s and 1930s, William Friedman started applying mathematics, statistics, and early electromechanical devices to cryptography,
- Mathematical version of Caesar cipher:

$$
\begin{aligned}
A &= 0 \\
B &= 1 \\
&\ldots \\
Z &= 25 \\
C_i &\equiv P_i + k \pmod{26}
\end{aligned}
$$

- Translation: the ith letter of ciphertext is produced by adding k to the ith letter of plaintext, and then taking the remainder after dividing by 26.

Information Theory

- Devised in 1948 by Claude Shannon.
- Provided a theoretical foundation for cryptography.
- Explained mathematically why knowing that “h” often follows ”t” (in English) helps solve ciphers: “h” has less *information*.
- Set the stage for modern cryptography and cryptanalysis.

What's a Cipher?

A cipher is a *function* that maps a *key* and *plaintext* to *ciphertext*, for which there is a corresponding decryption function:

$$\begin{aligned} c &= e(k,p) \\ p &= d(k,c) \end{aligned}$$

Put more formally,

$$\begin{aligned} E : K \times P &\mapsto C \\ D : K \times C &\mapsto P \end{aligned}$$

where K, P and C are sets.

Classically, P and C were the alphabet, though K wasn't. But they don't have to be!

Enter the Computer

- What is the obvious candidate for P and C on a computer?
- Bits? Bytes?
- Close — and sometimes right. But it's usually better to encrypt larger chunks.

Why Shouldn't We Encrypt Bytes?

- There are 256 possible bytes.
- For any given $k \in K$, the attacker only needs to compile a 256-element "codebook".
- In fact, given information theory, most of those entries will be very easy to build.
- We have to do better.

Let's Encrypt Larger Blocks

- The *Data Encryption Standard* (DES) encrypts 64-bit blocks, using 56-bit keys:

$$E : \{0,1\}^{56} \times \{0,1\}^{64} \mapsto \{0,1\}^{64}$$

- 2^{56} (72,057,594,037,927,936) possible keys.
- 2^{64} (18,446,744,073,709,551,616) code book entries for each key.
- (It turns out that even that's not enough.)

What is DES?

- In the early 1970s, the U.S. government issued an open call for an unclassified cipher for non-classified information.
- Eventually, IBM submitted a design called Lucifer.
- NSA tinkered with the design to produce DES.
- (There was a lot of suspicion and a lot of accusations that NSA tampered with the design to weaken it. Most of those accusations have since been proven false.)

How Does DES Work (Simplified)?

Repeat 16 times, for i ranging from 0 to 15:

Split the 64-bit block into two halves, L_i and R_i

$L_{i+1} = R_i$

$R_{i+1} = L_i \oplus f(R_i, K)$

Reassemble the two halves

Each round is easy to invert. But all 16 together are strong.

Encrypting Messages

- DES is a *block cipher*.
- What if we want to encrypt messages?
- Lots of ways; one is a *stream cipher*.
- Encryption function changes *state* each time, so that each encryption is different.

$$E : K \times P \times S \mapsto C \times S$$

- Example: use a counter. Encrypt the counter, add the result to the message, increment the counter by 1.
- Note: decryption needs the same starting counter.

Today's Block Cipher: AES

-
- The Advanced Encryption Standard is intended to replace DES.

⇒ Someone built a \$250,000 machine that could try all 2^{56} DES keys in a short time.

- New cipher is Rijndael, named after Joan Daemon and Vincent Rijmen.
- Encrypts 128-bit blocks.
- Key sizes of 128, 192, and 256 bits.

And Now for Something Completely Weird

In 1976, Whit Diffie and Marty Hellman had an insight: what if the encryption key k and the decryption key k *weren't* the same?

$$\begin{aligned} c &= e(k, p) \\ p &= d(k', c) \\ k &\neq k' \end{aligned}$$

Furthermore, it must be (for all practical purposes) impossible to find k' from k.

This was the root of *public key cryptography*.

What is Public Key Cryptography Good For?

- I publish my *public* (encryption) key in the phone book.
- You can use it to encrypt a message to me.
- I use my *private* (decryption) key to read it.

What's Wrong with that Scheme?

- Suppose I want to read email sent to you via public key cryptography.
- Further suppose that the "phone book" is really some Internet site.
- I hack the site and replace your public key with mine.
- I'll be able to read all your secret email.

Digital Signatures

- Diffie and Hellman had another insight.
- Suppose you *encrypted* a message with your secret *decryption* key.
- Only you know the decryption key, so only you can do that encryption.
- Everyone knows the public encryption key; anyone can use it to *decrypt* your message.
- This proves it came from you: a digital signature.

Certificates

- We can use digital signatures to defeat the attack.
- Assume that there is a mutually trusted party who has a private key S.
- This party uses S to sign a message containing my name and my public key:

$$d(\texttt{Steve Bellovin}||e_{SteveBellovin}, S)$$

- Such a construct is called a *certificate*.
- To use it, you first verify the signature against the third party's public key. Then you can extract my public key to send me a message.
- (Where do you get the trusted party's public key?)
- (What if someone hacks the trusted party's computer?)

Can Public Key Cryptography Exist?

- For digital signatures to exist, we need a *trapdoor function*: a function that's easy to calculate but extremely hard to invert.
- Diffie and Hellman couldn't quite invent one.
- But Ron Rivest, Adi Shamir, and Len Adleman succeeded.

RSA Encryption

- RSA encryption rests on two apparently-contradictory statements: It is relatively easy to tell if a large number is prime. But it is extremely hard to calculate the factors of a large composite number. Yes, that means that you don't do primality testing by lots of trial divisions.
- Pick two very large (hundreds of digits long) prime numbers, p and q; let $n = pq$.
- The public key is any number e, $1 < e < n$.
- The private key d is calculated by Euclid's algorithm such that

$$ed \equiv 1 \pmod{(p-1)(q-1)}$$

- Given only n and e, there is no way known to calculcate d without factoring n.

Encrypting with RSA

- To encrypt a message p with a public key of $\langle e, n \rangle$:

$$c = p^e \pmod{n}$$

- To decrypt:

$$p = c^d \pmod{n}$$

- This scheme works for digital signatures, too.
- In fact, you can sign before or after encrypting, to send a secret, signed message.
- N.B. The numbers in these modular exponentiations are hundreds of digits long. Public key cryptography is expensive...

How the World Learned of RSA

- Rivest, Shamir, and Adleman wrote a technical report.
- Martin Gardner described it in Scientific American.
- *Lots* of people requested copies of the report.
- Someone from NSA wrote to MIT, claiming that exporting the report violated the International Trafficking in Arms Regulations.
- Supposedly, this was a personal act, and not officially authorized. . .

The RSA Challenger

- Gardner's column gave a challenger cipher, using 100-bit primes.
- There's been progress in factoring since 1978.
- About 5 years ago, the message was decrypted:
  The magic words are squeamish ossifrage.

Cool Tricks: Coin Flipping

- How do we flip coins on the Internet? RSA lets us do it.
- Alice and Bob (A and B) each generate a public/private key pair E_A, D_A, E_B, D_B. (Both parties must use the same value for n.)
- Alice generates two random messages, M_h and M_t, for heads and tails, and sends $E_A(M_h)$ and $E_A(M_t)$ to Bob.
- Bob picks one of these messages, encrypts it, and sends back $E_B(E_A(M))$.
- Alice decrypts it and sends it back:

$$D_A(E_B(E_A(M)) = E_B(D_A(E_A(M))) = E_B(M)$$

- Bob decrypts this and gets either M_h or M_t, and sends it back to Alice.

Let's Try It

	e	d
Alice	5	53
Bob	13	61

$p = 7, q = 23, n = 161$

Assume $M_h = 14$.

$$
\begin{aligned}
14^5 \pmod{161} &= 84 \\
84^{13} \pmod{161} &= 28 \\
53^{28} \pmod{161} &= 126 \\
61^{126} \pmod{161} &= 14
\end{aligned}
$$

Other Cool Tricks

- Internet poker
- Simultaneous contract signing
- Secret-sharing
- Secure elections
- Digital cash

How is this Used Today?

- DES being replaced by AES.
- RSA is still believed secure, and is widely used on the Internet.
- Certificates are widely used; the public keys for major *certificate authorities* are built into browsers and operating systems.
- More and more Internet traffic is encrypted.

References

- *The Codebreakers*, David Kahn, Macmillan, 1967. The definitive work on the history of cryptography.
- *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*, Simon Singh, Anchor Books, 2000. More modern but less comprehensive than Kahn.
- *Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition*, Bruce Schneier, John Wiley & Sons, 1995. Detailed technical explanations of important cryptographic mechanisms.
- *Handbook of Applied Cryptography*, Alfred J. Menezes, Paul C. Van Oorschot, Scott A. Vanstone, CRC Press, 1996. A terse, comprehensive, highly mathematical treatment of cryptography. `http://www.cacr.math.uwaterloo.ca/hac`.

More References

- *The Design of Rijndael*, Joan Daemen and Vincent Rijmen, Springer, 2002.
- “New Directions in Cryptography”, Whitfield Diffie and Martin E. Hellman, *IEEE Transactions on Information Theory*, Nov. 1976. `http://www.cbcis.wustl.edu/~adpol/courses/cs502/Notes/diff.pdf`.
- “A Method of Obtaining Digital Signatures and Public-Key Cryptosystems”, Ronald L. Rivest, Adi Shamir, and Leonard Adleman, *Communications of the ACM*, Feb. 1978. `http://theory.lcs.mit.edu/~rivest/rsapaper.pdf`
- “A New Kind of Cipher That Would Take Millions of Years to Break”, Martin Gardner, *Scientific American*, Aug. 1977.

A Introduction to Modern Cryptography

smb@research.att.com

http://www.research.att.com/~smb

973-360-8656

AT&T Labs Research

Florham Park, NJ 07932