Computer Security

07. Cryptography

Paul Krzyzanowski
Rutgers University
Spring 2018
A secret manner of writing, … Generally, the art of writing or solving ciphers.

— *Oxford English Dictionary*
The analysis and decryption of encrypted text or information without prior knowledge of the keys.

— Oxford English Dictionary
1967 D. Kahn, *Codebreakers* p. xvi, Cryptology is the science that embraces cryptography and cryptanalysis, but the term ‘cryptology’ sometimes loosely designates the entire dual field of both rendering signals secure and extracting information from them.

— *Oxford English Dictionary*
Cryptography ≠ Security

Cryptography may be a component of a secure system

Just adding cryptography may not make a system secure
Cryptography: what is it good for?

- **Authentication**
  - determine origin of message

- **Integrity**
  - verify that message has not been modified

- **Nonrepudiation**
  - sender should not be able to falsely deny that a message was sent

- **Confidentiality**
  - others cannot read contents of the message
Terms

Plaintext (cleartext) message $P$

Encryption $E(P)$

Produces Ciphertext, $C = E(P)$

Decryption, $P = D(C)$

Cipher = cryptographic algorithm
Restricted cipher

Secret algorithm

- Vulnerable to:
  - Leaking
  - Reverse engineering
    - HD DVD (Dec 2006) and Blu-Ray (Jan 2007)
    - RC4
    - All digital cellular encryption algorithms
    - DVD and DIVX video compression
    - Firewire
    - Enigma cipher machine
    - Every NATO and Warsaw Pact algorithm during Cold War

- Hard to validate its effectiveness (who will test it?)
- Not a viable approach!
Shared algorithms & secret keys
BTW, the above is a bump key. See http://en.wikipedia.org/wiki/Lock_bumping.
The lock

Source: en.wikipedia.org/wiki/Pin_tumbler_lock
The key & lock

• We understand how the mechanism works:
  – Strengths
  – Weaknesses

• Based on this understanding, we can assess how much to trust the key & lock

Source: en.wikipedia.org/wiki/Pin_tumbler_lock
Kerckhoff’s Principle (1883)

A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.

Security should rest entirely on the secrecy of the key.
Properties of a good cryptosystem

• Ciphertext should be indistinguishable from random values

• Given ciphertext, there should be no way to extract the original plaintext or the key short of enumerating all possible keys (= brute force attack)

• The keys should be large enough that a brute force attack is not feasible
Symmetric key ciphers

Same secret key, $K$, for encryption & decryption

\[
C = E_K(P)
\]
\[
P = D_K(C)
\]
Classic Cryptosystems
Substitution Ciphers
Cæsar cipher

Earliest documented military use of cryptography
  – Julius Caesar c. 60 BC
  – shift cipher: simple variant of a substitution cipher
  – each letter replaced by one \( n \) positions away
    modulo alphabet size
    \( n = \) shift value = key

Similar scheme used in India
  – early Indians also used substitutions based on phonetics
    similar to pig latin

Last seen as ROT13 on Usenet to keep the reader from seeing offensive messages unwillingly
Cæsar cipher

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Cæsar cipher

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | |
### Cæsar cipher

**MY CAT HAS FLEAS**

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
Cæsar cipher

MY CAT HAS FLEAS

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
U V W X Y Z A B C D E F G H I J K L M N O P Q R S T
Cæsar cipher

MY CAT HAS FLEAS

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
U V W X Y Z A B C D E F G H I J K L M N O P Q R S T

GS
Caesar cipher

MY CAT HAS FLEAS

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

U V W X Y Z A B C D E F G H I J K L M N O P Q R S T

GSW
Cæsar cipher

MY CAT HAS FLEAS

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

U V W X Y Z A B C D E F G H I J K L M N O P Q R S T

GSWU
Cæsar cipher

MY CAT HAS FLEAS

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

U V W X Y Z A B C D E F G H I J K L M N O P Q R S T

GSWUN
Cæsar cipher

MY CAT HAS FLEAS

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
U V W X Y Z A B C D E F G H I J K L M N O P Q R S T

GSWUNB
Cæsar cipher

MY CAT HAS FLEAS

GSWUNBU
Cæsar cipher

MY CAT HAS FLEAS

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

U V W X Y Z A B C D E F G H I J K L M N O P Q R S T

GSWUNBUM
Cæsar cipher

MY CAT HAS FLEAS

GSWUNBUMZ
Cæsar cipher

MY CAT HAS FLEAS

GSWUNBUMZF
Cæsar cipher

MY CAT HAS FLEAS

ABCD E FGH IJKL MNOP QRST UVWX YZ
UVWXYZ ABCDE FGHIJKL MNOPQRST

GSWUNBUMZFY
Cæsar cipher

MY CAT HAS FLEAS

GSWUNBUMZFYU
Caesar cipher

MY CAT HAS FLEAS

GSWUNBMUFZYUM
Cæsar cipher

- MY CAT HAS FLEAS
- GSWUNBMUFZYM

- Convey one piece of information for decryption: *shift value*
- Trivially easy to crack
  (25 possibilities for a 26 character alphabet)
Ancient Hebrew variant (ATBASH)

MY CAT HAS FLEAS

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| Z | Y | X | W | V | U | T | S | R | Q | P | O | N | M | L | K | J | I | H | G | F | E | D | C | B | A |

NBXZGSZHUOVZH

- c. 600 BC
- No information (key) needs to be conveyed!
Monoalphabetic substitution cipher

MY CAT HAS FLEAS

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| M | P | S | R | L | Q | E | A | J | T | N | C | I | F | Z | W | O | Y | B | X | G | K | U | D | V | H |

IVSMXAMBQCLMB

- General case: arbitrary mapping
- both sides must have substitution alphabet
Monoalphabetic substitution cipher

Easy to decode:
– vulnerable to frequency analysis

<table>
<thead>
<tr>
<th>Moby Dick (1.2M chars)</th>
<th>Shakespeare (55.8M chars)</th>
</tr>
</thead>
<tbody>
<tr>
<td>e 12.300%</td>
<td>e 11.797%</td>
</tr>
<tr>
<td>o 7.282%</td>
<td>o 8.299%</td>
</tr>
<tr>
<td>d 4.015%</td>
<td>d 3.943%</td>
</tr>
<tr>
<td>b 1.773%</td>
<td>b 1.634%</td>
</tr>
<tr>
<td>x 0.108%</td>
<td>x 0.140%</td>
</tr>
</tbody>
</table>
Statistical Analysis

Letter frequencies

- E: 12%
- A, H, I, N, O, R, S, T: 6 – 9%
- D, L: 4%
- B, C, F, G, M, P, U, W, Y: 1.5 – 2.8%
- J, K, Q, V, X, Z: < 1%

Common digrams:

- TH (3.56%), HE (3.07%), IN (2.43%), ER (2.05%), AN, RE, …

Common trigrams

- THE, ING, AND, HER, ERE, …
Polyalphabetic substitution ciphers

• Designed to thwart frequency analysis techniques
  – different ciphertext symbols can represent the same plaintext symbol
    • $1 \rightarrow \textit{many}$ relationship between letter and substitute

• Leon Battista Alberti: 1466: invented the \textit{key}
  – two disks
  – line up predetermined letter on inner disk with outer disk
  – plaintext on inner $\rightarrow$ ciphertext on outer
  – after $n$ symbols, the disk is rotated to a new alignment

encrypt: $A \rightarrow J$
decrypt: $J \rightarrow A$
Vigenère polyalphabetic cipher

Blaise de Vigenère, court of Henry III of France, 1518

- Use table and key word to encipher a message
- repeat keyword over text: (e.g. key=FACE)
  
  FA CEF ACE FACEF ....
  MY CAT HAS FLEAS

- **encrypt**: find intersection:
  
  row = keyword letter
  column = plaintext letter

- **decrypt**: column = keyword letter, search for intersection
  = ciphertext letter

- message is encrypted with as many substitution ciphers as there are letters in the keyword
### Vigenère polyalphabetic cipher

The Vigenère cipher is a method of encrypting alphabetic text by using a series of interwoven Caesar ciphers based on the letters of a keyword. It is a polyalphabetic substitution cipher.

#### Key Parts
- **Plaintext Letter**: The original text letter to be encrypted.
- **Keytext Letter**: The letter of the keyword that is used to find the corresponding ciphertext letter in the Vigenère table.
- **Ciphertext Letter**: The letter that results from encrypting the plaintext letter using the keytext letter.

#### Diagram

The diagram shows a Vigenère table with interwoven Caesar ciphers. The table is a grid of the alphabet, where each row represents a Caesar cipher. To encrypt a plaintext letter, you find the corresponding position in the table based on the keytext letter. In the example, the keytext letter M corresponds to the ciphertext letter S.

#### Example

Let's encrypt the word **hello** using the keyword **world**.

**Plaintext**: hello
**Keytext**: world

**Ciphertext**: bgnns

**Steps**:
1. **h** + **w** = **b**
2. **e** + **o** = **g**
3. **l** + **r** = **n**
4. **o** + **l** = **n**
5. **l** + **d** = **s**

Thus, the ciphertext is **bgnns**.
Vigenère polyalphabetic cipher

FA CEF ACE FACEF
MY CAT HAS FLEAS

R
Vigenère polyalphabetic cipher

FA CEF ACE FACEF
MY CAT HAS FLEAS

RY
Vigenère polyalphabetic cipher

FA CEF ACE FACEF
MY CAT HAS FLEAS

RY E

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
Vigenère polyalphabetic cipher

FA CEF ACE FACEF
MY CAT HAS FLEAS

RY EE
Vigenère polyalphabetic cipher

FA CEF ACE FACEF
MY CAT HAS FLEAS

RY EEY
Vigenère polyalphabetic cipher

FA CEF ACE FACEF
MY CAT HAS FLEAS

RY EEY H

|   | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| A |   |   |   |   |   |   |   |   | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B |   |   |   |   |   |   |   |   | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| D |   |   |   |   |   |   |   |   |   |   | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E |   |   |   |   |   |   |   |   |   |   |   | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F |   |   |   |   |   |   |   |   |   |   |   |   | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G |   |   |   |   |   |   |   |   |   |   |   |   |   | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |

MY CAT HAS FLEAS

FA CEF ACE FACEF

RY EEY H

March 20, 2018
CS 419 © 2018 Paul Krzyzanowski
50
Vigenère polyalphabetic cipher

FA CEF ACE FACEF
MY CAT HAS FLEAS

RY EEY HC
### Vigenère polyalphabetic cipher

**FA CEF ACE FACEF**

**MY CAT HAS FLEAS**

**RY EEY HCW**

```
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
```

**MY CAT HAS FLEAS**

**RY EEY HCW**
Vigenère polyalphabetic cipher

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
Vigenère polyalphabetic cipher

FA CEF ACE FACEF
MY CAT HAS FLEAS

RY EEY HCW KL

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |   |
| C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |   |
| D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |   |
| E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |   |
| F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |   |
| G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |   |
| H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |   |
Vigenère polyalphabetic cipher

```
MY CAT HAS FLEAS
_____________
RY EEY HCW KLG
```

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |

March 20, 2018

CS 419 © 2018 Paul Krzyzanowski
Vigenère polyalphabetic cipher

FA CEF ACE FACEF
MY CAT HAS FLEAS
_____________
RY EEY HCW KLGE
Vigenère polyalphabetic cipher

FA CEF ACE FACEF
MY CAT HAS FLEAS

RY EEY HCW KLGEX

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |

March 20, 2018
CS 419 © 2018 Paul Krzyzanowski
"The rebels reposed their major trust, however, in the Vigenère, sometimes using it in the form of a brass cipher disc. In theory, it was an excellent choice, for so far as the South knew the cipher was unbreakable. In practice, it proved a dismal failure. For one thing, transmission errors that added or subtracted a letter ... unmeshed the key from the cipher and caused no end of difficulty. Once Major Cunningham of General Kirby-Smith's staff tried for twelve hours to decipher a garbled message; he finally gave up in disgust and galloped around the Union flank to the sender to find out what it said."

http://rz1.razorpoint.com/index.html
Cryptoanalysis of the Vigenére cipher

• Hard to break with long keys and small amounts of ciphertext
  ... in the 1800s

• Cryptoanalysis of the Vigenére cipher
  1. Determine key length
     • Count coincidences – identical characters $n$ characters apart
     • Find high values of $n$ – that tells you the period of the key
  2. Determine values of each character of the key
     • You know the length of the key – that’s the # of Caesar ciphers you have
     • Do a frequency analysis of each position of the key.
One-time pad

Only provably secure encryption scheme

• Invented in 1917

• Large non-repeating set of random key letters originally written on a pad

• Each key letter on the pad encrypts exactly one plaintext character
  – Encryption is addition of characters modulo 26

• Sender destroys pages that have been used

• Receiver maintains identical pad
One-time pad

If pad contains

KWXOPWMAELGHW…

and we want to encrypt

MY CAT HAS FLEAS

Ciphertext =

WUZOIDMSJWKHO

M + K mod 26 = W
Y + W mod 26 = U
C + X mod 26 = Z
A + O mod 26 = O
T + P mod 26 = I
H + W mod 26 = D
A + M mod 26 = M
S + A mod 26 = S
F + E mod 26 = J
L + L mod 26 = W
E + G mod 26 = K
A + H mod 26 = H
S + W mod 26 = O
One-time pad

The same ciphertext can decrypt to *anything* depending on the key!

Same ciphertext:

```
WUZOIDMSJWKHO
```

With a pad containing:

```
KWXOPWMAELGHW...
```

Produces:

```
THE DOG IS HAPPY
```
One-time pads in computers

Can be extended to binary data

- Random key sequence as long as the message
- Exclusive-or key sequence with message
- Receiver has the same key sequence
void onetimepad(void)
{
    FILE *if = fopen("intext", "r");
    FILE *kf = fopen("keytext", "r");
    FILE *of = fopen("outtext", "w");
    int c, k;

    while ((c = getc(if)) != EOF) {
        k = getc(kf);
        putc((c^k), of);
    }
    fclose(if); fclose(kf); fclose(of);
}
One-time pads: perfect secrecy

Perfect secrecy

– Ciphertext conveys no information about the content of plaintext
– *Achieved only if there are as many possible keys as plaintext*

Problems with one-time pads:

– **Key needs to be as long as the message!**
– Key storage can be problematic
  • May need to store a lot of data
– Keys have to be generated randomly
  • Cannot use pseudo-random number generator
– Cannot reuse key sequence
– Sender and receiver *must* remain synchronized
  (e.g. cannot lose a message)
Random numbers

“Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin”

– John vonNeumann

• Pseudo-random generators
  – Linear feedback shift registers
  – Multiplicative lagged Fibonacci generators
  – Linear congruential generator

• Obtain randomness from:
  – Time between keystrokes
  – Various network/kernel events
  – Cosmic rays
  – Electrical noise
  – Other encrypted messages
Stream ciphers

Key stream generator produces a sequence of pseudo-random bytes

\[ C_i = S_i \oplus P_i \]
Stream ciphers

Can never reuse a key

\[ C = A \oplus K \]

\[ C' = B \oplus K \]

\[ C \oplus C' = A \oplus K \oplus B \oplus K = A \oplus B \]

Guess A and see if B makes sense
Electro-mechanical cryptographic engines
Rotor machines

1920s: mechanical devices used for automating encryption

Rotor machine:

– Set of independently rotating cylinders (rotors) through which electrical pulses flow

– Each rotor has input & output pin for each letter of the alphabet
  • Each rotor implements a substitution cipher

– Output of each rotor is fed into the next rotor

– Together they implement a version of the Vigenère cipher
Rotor machines

• Simplest rotor machine: single cylinder

After a character is entered, the cylinder rotates one position
- internal combinations shifted by one
- polyalphabetic substitution cipher with a period of 26
Single cylinder rotor machine
Single cylinder rotor machine

MY CAT HAS FLEAS

S
MY CAT HAS FLEAS

SU
MY CAT HAS FLEAS

SUI
Single cylinder rotor machine

MY CAT HAS FLEAS

SUIU
Single cylinder rotor machine

MY CAT HAS FLEAS

SUIUV
Single cylinder rotor machine

MY CAT HAS FLEAS

SUIUVA
MY CAT HAS FLEAS

SUIUVAY
Single cylinder rotor machine

MY CAT HAS FLEAS

SUIUVAYO
Single cylinder rotor machine

MY CAT HAS FLEAS

SUIUVAYOI
Single cylinder rotor machine

MY CAT HAS FLEAS

SUIUVAYOIN
Single cylinder rotor machine

MY CAT HAS FLEAS

SUIUVAYOINK
Single cylinder rotor machine

MY CAT HAS FLEAS

SUIUVAYOINKB
Single cylinder rotor machine

MY CAT HAS FLEAS

SUIUVAYOINKBY
Multi-cylinder rotor machines

Single cylinder rotor machine
  – Substitution cipher with a period = length of alphabet (e.g., 26)

Multi-cylinder rotor machine
  – Feed output of one cylinder as input to the next one
  – First rotor advances after character is entered
  – Second rotor advances after a full period of the first

Polyalphabetic substitution cipher
  • Period = \((\text{length of alphabet})^{\text{number of rotors}}\)
  • 3 26-char cylinders \(\Rightarrow 26^3 = 17,576\) substitution alphabets
  • 5 26-char cylinders \(\Rightarrow 26^5 = 11,881,367\) substitution alphabets
Enigma

• Enigma machine used in Germany during WWII

• Three rotor system
  – $26^3 = 17,576$ possible rotor positions

• Input data permuted via patch panel before sending to rotor engine

• Data from last rotor reflected back through rotors
  ⇒ *makes encryption symmetric*

• Need to know initial settings of rotor
  – setting was $f(date)$ in a book of codes

• Broken by group at Bletchley Park (Alan Turing)
Enigma

Reflector

Rotors

Plugboard

Glowlamps (results)

Keyboard (input)
Transposition Ciphers
Transposition ciphers

• Permute letters in plaintext according to rules

• Knowledge of rules will allow message to be decrypted

• First mentioned in Greece in the 7th century BC
  – Skytale (rhymes with Italy) = staff cipher
Transposition ciphers: skytale

Secret = diameter of skytale
Transposition ciphers: skytale

MYCATHASFLEAS

MYC
HAS
EAS

Y
A
A

MHEYAA
Transposition ciphers: skytale

MYCATHASFLEAS

MHEYAACSS
Transposition ciphers: skytale

MYCATHASFLEAS

CAT
SFL
Sxy

MHEYAACSSAFx

Pad out the text
Transposition ciphers: skytale
Skytale as a set of columns

Table version of skytale
- enter data horizontally, read it vertically
- secrecy is the width of the table

MYCATHASFLEAS

MYCA
THAS
FLEA
Sxyz
Skytale as a set of columns

Table version of skytale
- enter data horizontally, read it vertically
- secrecy is the width of the table

<table>
<thead>
<tr>
<th>MYCATHASFLEAS</th>
<th>M T F S</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Y C A</td>
</tr>
<tr>
<td></td>
<td>H A S</td>
</tr>
<tr>
<td></td>
<td>L E A</td>
</tr>
<tr>
<td></td>
<td>x y z</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>M T F S</td>
</tr>
</tbody>
</table>

March 20, 2018
CS 419 © 2018 Paul Krzyzanowski
Skytale as a set of columns

Table version of skytale
– enter data horizontally, read it vertically
– secrecy is the width of the table
Skytale as a set of columns

Table version of skytale

– enter data horizontally, read it vertically
– secrecy is the width of the table

MYCATHASFLEAS → MYCATHASFLEAS → MTFSYHLxCAEy
Skytale as a set of columns

Table version of skytale

– enter data horizontally, read it vertically
– secrecy is the width of the table

MYCATHASFLEAS → MYCAT → A → MYCATHASFLEAS

MTFSYHLxCAEyASAz
Columnar transposition cipher

- Permute letters in plaintext according to key
- Read down columns, sorting by key

Key: \[ 3 \ 1 \ 4 \ 2 \]

MYCATHASFLEAS

\[
\begin{array}{cccc}
M & Y & C & A \\
T & H & A & S \\
F & L & E & A \\
S & x & y & z \\
\end{array}
\]
Columnar transposition cipher

- Permute letters in plaintext according to key
- Read down columns, sorting by key

Key:

<table>
<thead>
<tr>
<th>3</th>
<th>1</th>
<th>4</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Y</td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>T</td>
<td>H</td>
<td>A</td>
<td>S</td>
</tr>
<tr>
<td>F</td>
<td>L</td>
<td>E</td>
<td>A</td>
</tr>
<tr>
<td>S</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
</tbody>
</table>

MYCATHASFLEAS → YHLx

YHLx
Columnar transposition cipher

- Permute letters in plaintext according to **key**
- Read down columns, sorting by key

Key:

```
3 1 4 2
M Y C A
T H A S
F L E A
S x y z
```

MYCATHASFLEAS

```
3
M Y C
1
T H A
4
F L E
```

```
M Y C A
T H A S
F L E A
S x y z
```

```
YHLxASAz
```

ASAz
Columnar transposition cipher

- Permute letters in plaintext according to **key**
- Read down columns, sorting by key

Key:

```
3 1 4 2
M Y C A
T H A S
F L E A
S x y z
```

MYCATHASFLEAS → YHLxASAxMTFS

MTFS
Columnar transposition cipher

- Permute letters in plaintext according to **key**
- Read down columns, sorting by key

Key: 3 1 4 2

MYCATHASFLEAS

YHLxASAzMTFSCAEy

CAEY
Columnar transposition cipher

- Permute letters in plaintext according to **key**
- Read down columns, sorting by key

**Key:**

```
3  1  4  2
M  Y  C  A
T  H  A  S
F  L  E  A
S  x  y  z
```

MYCATHASFLEAS → YHLxASAzMTFSCAEy
Transposition cipher

- Not vulnerable to frequency analysis

- Scytale trivial to attack
  - Make all possible matrices that would fit the ciphertext
  - Write ciphertext across rows
  - See if the columns contain legible content

- Scrambled columns make it a bit harder
  - Need to permute columns of matrices
Combined ciphers

• Combine transposition with substitution ciphers
  – German ADFGVX cipher (WWI)

• Can be troublesome to implement
  – Requires memory
  – Requires block processing (these are block ciphers)

• Difficult with manual cryptography
Computer Cryptography
Block ciphers

- Block ciphers were a pain without computers
  ... but dominate computer cryptography

Encrypt a fixed number of bits at a time
Output blocksize (usually) = input blocksize
Structure of block ciphers

• Multiple **rounds** of combining the plaintext with the key

• Optional:
  – Convert key to internal form (possibly different per round)

• DES: 16 rounds

• AES: 10-14 rounds, depending on key length

Sounds easy … but is difficult to design
DES

- Data Encryption Standard
  - Adopted as a federal standard in 1976
- Block cipher, 64 bit blocks, 56 bit key
- Substitution followed by a permutation
  - Transposition and XORs based on subkey derived from the key
  - 16 rounds
DES

64 bit plaintext block

initial permutation, IP

left half, L₁

right half, R₀

16 rounds

L₁ = R₀

R₁ = L₀ ⊕ f(R₀, K₁)

L₁₅ = R₁₄

R₁₅ = L₁₄ ⊕ f(R₁₄, K₁₅)

R₁₆ = L₁₅ ⊕ f(R₁₅, K₁₆)

L₁₆ = R₁₅

final permutation, IP⁻¹

64 bit ciphertext block

48-bit subkey permuted from key

K₁

K₁₆
DES: \( f \) per round

DATA: right 32 bits

KEY: 56 bits

DATA: left 32 bits

New DATA: right 32 bits
DES: S-boxes

• After compressed key is XORed with expanded block
  – 48-bit result moves to substitution operation via eight substitution boxes (s-boxes)

• Each S-box has
  – 6-bit input
  – 4-bit output

• 48 bits divided into eight 6-bit sub-blocks

• Each block is operated by a separate S-box

• S-boxes are key components of DES’s security

• Net result: 48 bit input generates 32 bit output
Is DES secure?

• 56-bit key makes DES relatively weak
  – $2^{56} = 7.2 \times 10^{16}$ keys
  – Brute-force attack

• By the late 1990’s:
  – DES cracker machines built to crack DES keys in a few hours
  – DES Deep Crack: 90 billion keys/second
  – Distributed.net: test 250 billion keys/second

• Now you can build a DES cracker for < $10,000
The power of 2

- Adding one extra bit to a key doubles the search space.
- Suppose it takes 1 second to search through all keys with a 20-bit key.

<table>
<thead>
<tr>
<th>key length</th>
<th>number of keys</th>
<th>search time</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 bits</td>
<td>1,048,576</td>
<td>1 second</td>
</tr>
<tr>
<td>21 bits</td>
<td>2,097,152</td>
<td>2 seconds</td>
</tr>
<tr>
<td>32 bits</td>
<td>$4.3 \times 10^9$</td>
<td>~ 1 hour</td>
</tr>
<tr>
<td>56 bits</td>
<td>$7.2 \times 10^{16}$</td>
<td>2,178 years</td>
</tr>
<tr>
<td>64 bits</td>
<td>$1.8 \times 10^{19}$</td>
<td>&gt; 557,000 years</td>
</tr>
<tr>
<td>256 bits</td>
<td>$1.2 \times 10^{77}$</td>
<td>$3.5 \times 10^{63}$ years</td>
</tr>
</tbody>
</table>

Distributed & custom hardware efforts typically allow us to search between 1 and >100 billion 64-bit (e.g., RC5) keys per second.
Increasing The Key

Can double encryption work for DES?

- Useless if we could find a key $K$ such that:

$$E_K(P) = E_{K_2}(E_{K_1}(P))$$

- This does not hold for DES (luckily!)
Double DES

Vulnerable to meet-in-the-middle attack

If we know some pair (P, C), then:

[1] Encrypt P for all $2^{56}$ values of $K_1$
[2] Decrypt C for all $2^{56}$ values of $K_2$

For each match where [1] = [2]

– Test the two keys against another P, C pair
– If match, you are assured that you have the key
Triple DES key lengths

Triple DES with two 56-bit keys (112-bit key):

\[ C = E_{K_1}(D_{K_2}(E_{K_1}(P))) \]

Triple DES with three 56-bit keys (168-bit key):

\[ C = E_{K_3}(D_{K_2}(E_{K_1}(P))) \]

Decryption used in middle step for compatibility with DES (\(K_1=K_2=K_3\))

\[ C = E_K(D_K(E_K(P))) \equiv C = E_{K_1}(P) \]
AES … successor to DES

From NIST:

Assuming that one could build a machine that could recover a DES key in a second (i.e., try $2^{56}$ keys per second), then it would take that machine approximately 149 trillion years to crack a 128-bit AES key. To put that into perspective, the universe is believed to be less than 20 billion years old.

http://csrc.nist.gov/encryption/aes/
The End