Network Security

<CH 2>

Youn Kyu Lee
Hongik University

Crypto

- Cryptography – the science and art of designing ciphers (making 'secret codes')
- Cryptanalysis – the science and art of breaking them (breaking 'secret codes')
- Cryptology – the study of both
- Crypto – all of the above (context-aware)

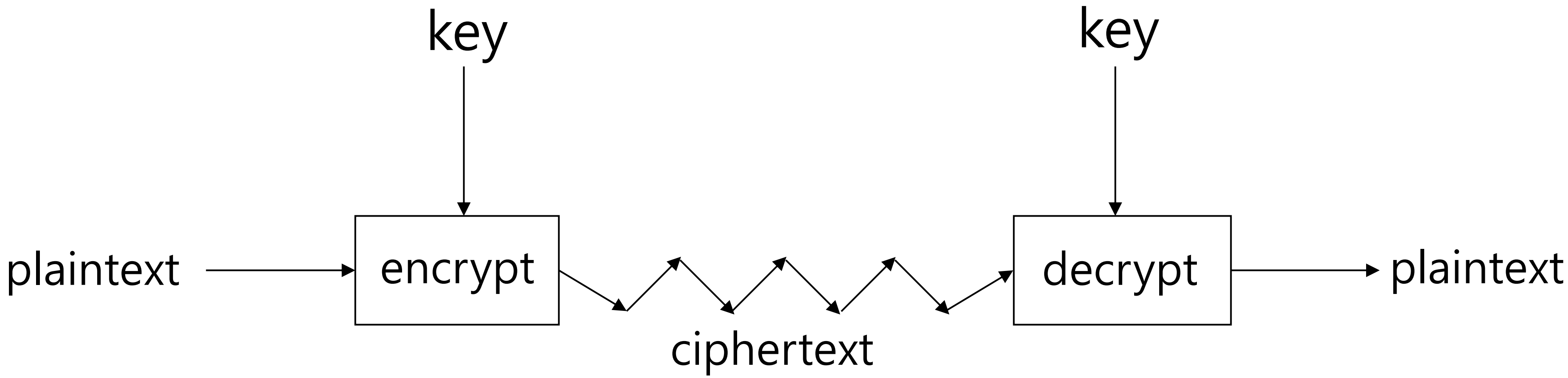
Basic Definitions

- A *cipher* or *cryptosystem* is used to *encrypt* the *plaintext(cleartext)*
- The result of encryption is *ciphertext*
- We *decrypt* ciphertext to recover plaintext
- Some building blocks: block ciphers, stream ciphers, hash functions
- Or key-based classifications: 1 key, 2 keys, …
- A *key* is used to configure a cryptosystem
- A *symmetric key* cryptosystem uses the same key to encrypt as to decrypt
- A *public key(asymmetric key)* cryptosystem uses a *public key* to encrypt and a *private key* to decrypt

Crypto

- Basic assumptions
  - The system is completely known to the attacker
  - Only the key is secret
  - That is, crypto algorithms are not secret
- Why do we make this assumption?
  - Experience has shown that secret algorithms are weak when exposed
  - Secret algorithms never remain secret
  - Better to find weaknesses beforehand

Crypto as Black Box



A generic view of symmetric key crypto

Historic Example of Simple ciphers

Shift Cipher: Treat letters $\{A, \ldots, Z\}$ like integers $\{0, \ldots, 25\} = \mathbb{Z}_{26}$. Chose *key* $K \in \mathbb{Z}_{26}$, *encrypt* by addition modulo 26, *decrypt* by subtraction modulo 26.

Example with K=25: `IBM`$\rightarrow$`HAL`.

With $K = 3$ known as *Caesar Cipher*, with $K = 13$ known as `rot13`.

The tiny key space size 26 makes *brute force* key search trivial.

Transposition Cipher: K is permutation of letter positions.

Key space is $n!$, where n is the permutation block length.

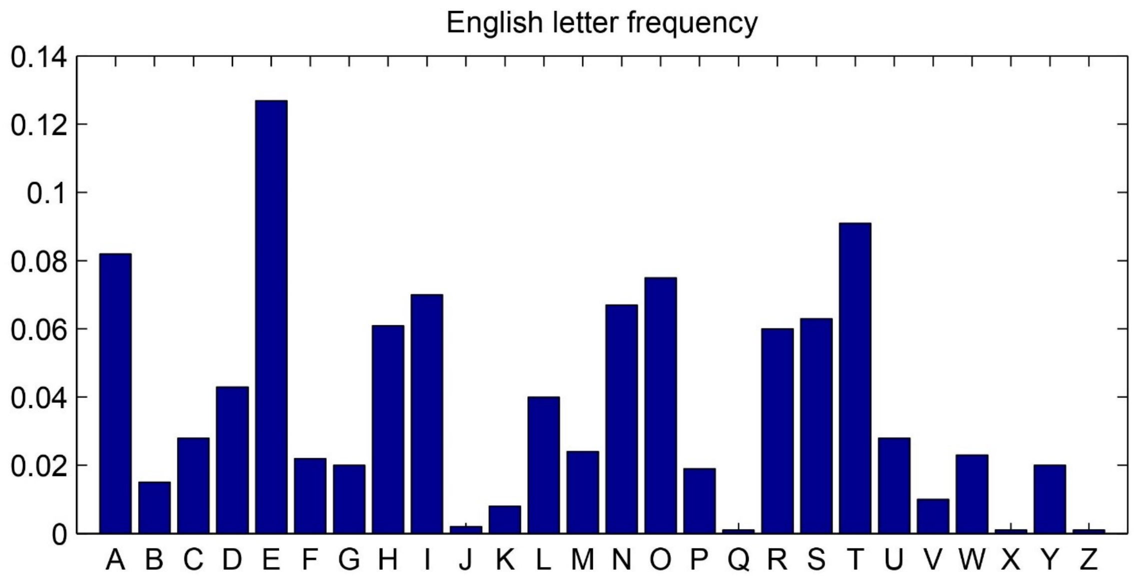
Substitution Cipher (monoalphabetic): Key is permutation $K : \mathbb{Z}_{26} \leftrightarrow \mathbb{Z}_{26}$. Encrypt plaintext $P = p_1 p_2 \ldots p_m$ with $c_i = K(p_i)$ to get ciphertext $C = c_1 c_2 \ldots c_m$, decrypt with $p_i = K^{-1}(c_i)$.

Key space size $26! > 4 \times 10^{26}$ makes brute force search infeasible.

Monoalphabetic substitution ciphers allow easy ciphertext-only attack with the help of language statistics (for messages that are at least few hundred characters long):

from Introduction to Security by Markus Kuhn
http://www.cl.cam.ac.uk/teaching/0708/IntroSec/

Historic Example of Simple ciphers



The most common letters in English:
E, T, A, O, I, N, S, H, R, D, L, C, U, M, W, F, G, Y, P, B, V, K, . . .

The most common digrams in English:
TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, . . .

The most common trigrams in English:
THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR, . . .

from Introduction to Security by Markus Kuhn
http://www.cl.cam.ac.uk/teaching/0708/IntroSec/

Historic Example of Simple ciphers

Vigenère cipher

Inputs:

- $\longrightarrow$ Key word $K = k_1 k_2 \ldots k_n$
- $\longrightarrow$ Plain text $P = p_1 p_2 \ldots p_m$

Encrypt into ciphertext:

$$c_i = (p_i + k_{[(i-1) \bmod n]+1}) \bmod 26.$$

Example: $K = \mathrm{SECRET}$

S	E	C	R	E	T	S	E	C	...
A	T	T	A	C	K	A	T	D	...
S	X	V	R	G	D	S	X	F	...

The modular addition can also be replaced with XOR or any other group operator available on the alphabet. Vigenère is an example of a *polyalphabetic* cipher.

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
BCDEFGHIJKLMNOPQRSTUVWXYZA
CDEFGHIJKLMNOPQRSTUVWXYZAB
DEFGHIJKLMNOPQRSTUVWXYZABC
EFGHIJKLMNOPQRSTUVWXYZABCD
FGHIJKLMNOPQRSTUVWXYZABCDE
GHIJKLMNOPQRSTUVWXYZABCDEF
HIJKLMNOPQRSTUVWXYZABCDEFG
IJKLMNOPQRSTUVWXYZABCDEFGH
JKLMNOPQRSTUVWXYZABCDEFGHI
KLMNOPQRSTUVWXYZABCDEFGHIJ
LMNOPQRSTUVWXYZABCDEFGHIJK
MNOPQRSTUVWXYZABCDEFGHIJKL
NOPQRSTUVWXYZABCDEFGHIJKLM
OPQRSTUVWXYZABCDEFGHIJKLMN
PQRSTUVWXYZABCDEFGHIJKLMNO
QRSTUVWXYZABCDEFGHIJKLMNOP
RSTUVWXYZABCDEFGHIJKLMNOPQ
STUVWXYZABCDEFGHIJKLMNOPQR
TUVWXYZABCDEFGHIJKLMNOPQRS
UVWXYZABCDEFGHIJKLMNOPQRST
VWXYZABCDEFGHIJKLMNOPQRSTU
WXYZABCDEFGHIJKLMNOPQRSTUV
XYZABCDEFGHIJKLMNOPQRSTUVW
YZABCDEFGHIJKLMNOPQRSTUVWX
ZABCDEFGHIJKLMNOPQRSTUVWXY
```

from Introduction to Security by Markus Kuhn
http://www.cl.cam.ac.uk/teaching/0708/IntroSec/

Historic Example of Simple ciphers

Perfect secrecy

→ Computational security – The most efficient known algorithm for breaking a cipher would require far more computational steps than any hardware available to an opponent can perform.

→ Unconditional security – The opponent has not enough information to decide whether one plaintext is more likely to be correct than another, even if unlimited computational power were available.

Perfect secrecy means that the cryptanalyst's a-posteriori probability distribution of the plaintext, after having seen the ciphertext, is identical to its a-priori distribution. In other words: looking at the ciphertext leads to no new information.

C.E. Shannon: Communication theory of secrecy systems. Bell System Technical Journal, Vol 28, Oct 1949, pp 656–715. `http://netlab.cs.ucla.edu/wiki/files/shannon1949.pdf`

from Introduction to Security by Markus Kuhn
http://www.cl.cam.ac.uk/teaching/0708/IntroSec/

Historic Example of Simple ciphers

Vernam cipher / one-time pad

The one-time pad is a variant of the Vigenère Cipher with $m = n$. The key is as long as the plaintext, and no key letter is ever used to encrypt more than one plaintext letter.

For each possible plaintext P, there exists a key K that turns a given ciphertext C into $P = D(K, C)$. If all K are equally likely, then also all P will be equally likely for a given C, which fulfills Shannon's definition of perfect secrecy.

One-time pads have been used intensively during significant parts of the 20th century for diplomatic communications security, e.g. on the telex line between Moscow and Washington. Keys were generated by hardware random bit stream generators and distributed via trusted couriers.

In the 1940s, the Soviet Union encrypted part of its diplomatic communication using recycled one-time pads, leading to the success of the US decryption project VENONA.
`http://www.nsa.gov/venona/`

from Introduction to Security by Markus Kuhn
http://www.cl.cam.ac.uk/teaching/0708/IntroSec/

One-Time Pad: Encryption

```
e=000   h=001   i=010   k=011   l=100   r=101   s=110   t=111
```

Encryption: Plaintext ⊕ Key = Ciphertext

	h	e	i	l	h	i	t	l	e	r
Plaintext:	001	000	010	100	001	010	111	100	000	101
Key:	111	101	110	101	111	100	000	101	110	000
Ciphertext:	110	101	100	001	110	110	111	001	110	101
	s	r	l	h	s	s	t	h	s	r

One-Time Pad: Decryption

`e=000   h=001   i=010   k=011   l=100   r=101   s=110   t=111`

Decryption: Ciphertext ⊕ Key = Plaintext

	s	r	l	h	s	s	t	h	s	r
Ciphertext:	110	101	100	001	110	110	111	001	110	101
Key:	111	101	110	101	111	100	000	101	110	000
Plaintext:	001	000	010	100	001	010	111	100	000	101
	h	e	i	l	h	i	t	l	e	r

One-Time Pad

Double agent claims sender used following "**key**"

	s	r	l	h	s	s	t	h	s	r
Ciphertext:	110	101	100	001	110	110	111	001	110	101
"**key**":	101	111	000	101	111	100	000	101	110	000
"Plaintext":	011	010	100	100	001	010	111	100	000	101
	k	i	l	l	h	i	t	l	e	r

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

One-Time Pad Summary

- **Provably secure**… (= **unconditionally secure**)

  (= ensuring **Perfect Secrecy)**
  - Ciphertext provides **no** information about plaintext
  - All plaintexts are equally likely
- …but, only when be used correctly
  - Pad must be random, used only once
  - Pad is known only to sender and receiver
- Note: pad (key) is same size as message
- So, why not distribute msg instead of pad?

Cryptanalysis: Terminology

- Cryptosystem is **secure** if best known attack is to try all keys
  - that is, Exhaustive key search
- Cryptosystem is **insecure** if ***any*** shortcut attack is known

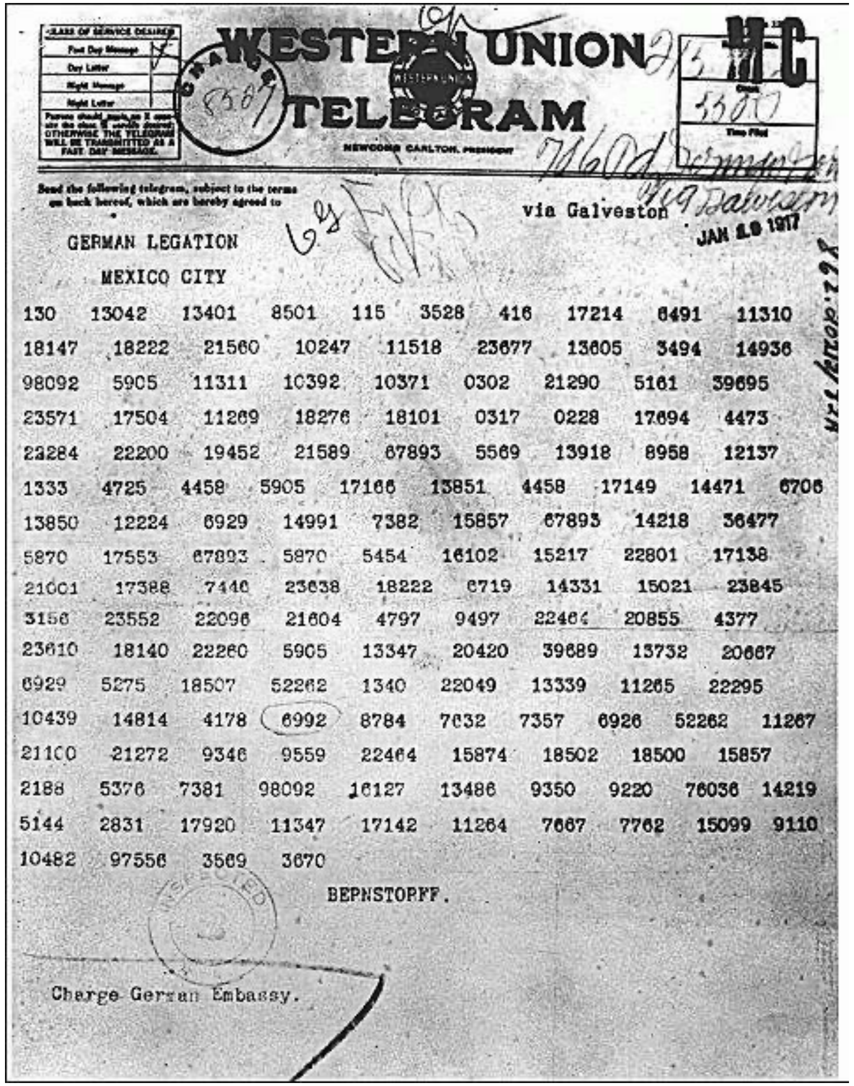
Codebook Cipher

- Literally, a book filled with “codewords”
- Zimmerman Telegram encrypted via codebook

Februar	13605
fest	13732
finanzielle	13850
folgender	13918
Frieden	17142
Friedenschluss	17149
:	:

Zimmerman Telegram

- Perhaps most famous codebook ciphertext ever
- A major factor in U.S. entry into World War I


WESTERN UNION TELEGRAM

NEWCOMB CARLTON, PRESIDENT

Send the following telegram, subject to the terms on back hereof, which are hereby agreed to

via Galveston

JAN 18 1917

GERMAN LEGATION

MEXICO CITY

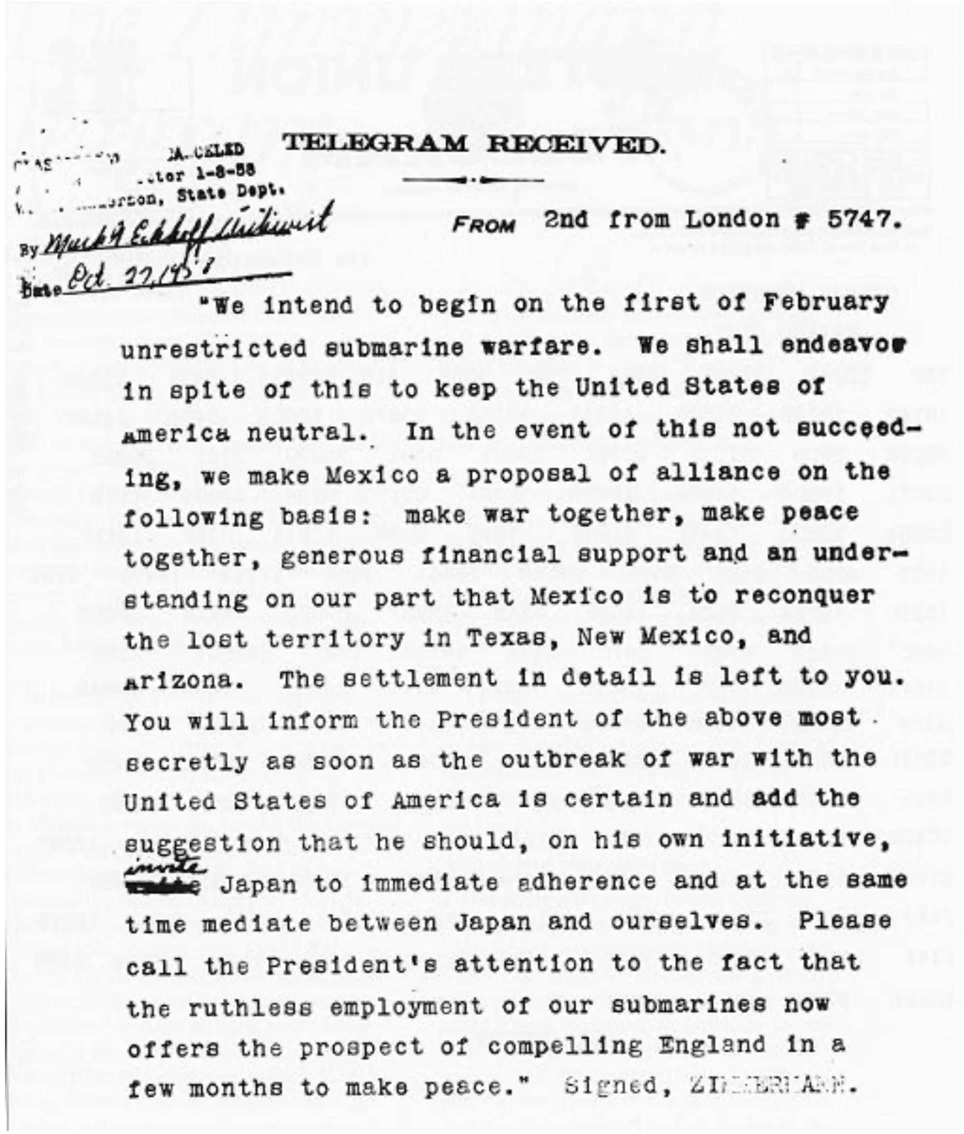
130 13042 13401 8501 115 3528 416 17214 6491 11310
18147 18222 21560 10247 11518 23677 13605 3494 14936
98092 5905 11311 10392 10371 0302 21290 5161 39695
23571 17504 11269 18276 18101 0317 0228 17694 4473
22284 22200 19452 21589 67893 5569 13918 8958 12137
1333 4725 4458 5905 17166 13851 4458 17149 14471 6706
13850 12224 6929 14991 7382 15857 67893 14218 36477
5870 17553 67893 5870 5454 16102 15217 22801 17138
21001 17388 7446 23638 18222 6719 14331 15021 23845
3156 23552 22096 21604 4797 9497 22464 20855 4377
23610 18140 22260 5905 13347 20420 39689 13732 20667
6929 5275 18507 52262 1340 22049 13339 11265 22295
10439 14814 4178 6992 8784 7632 7357 6926 52262 11267
21100 21272 9346 9559 22464 15874 18502 18500 15857
2188 5376 7381 98092 16127 13486 9350 9220 76036 14219
5144 2831 17920 11347 17142 11264 7667 7762 15099 9110
10482 97556 3569 3670

BEPNSTOPFF.

Charge German Embassy.

Zimmerman Telegram Decrypted

- British had recovered partial cod ebook
- Then able to fill in missing parts
- Modern **block ciphers** are codebooks!



TELEGRAM RECEIVED.

By *Mark A. Eckhoff*

Date *Oct. 22, 1957*

FROM 2nd from London # 5747.

"We intend to begin on the first of February unrestricted submarine warfare. We shall endeavor in spite of this to keep the United States of America neutral. In the event of this not succeeding, we make Mexico a proposal of alliance on the following basis: make war together, make peace together, generous financial support and an understanding on our part that Mexico is to reconquer the lost territory in Texas, New Mexico, and Arizona. The settlement in detail is left to you. You will inform the President of the above most secretly as soon as the outbreak of war with the United States of America is certain and add the suggestion that he should, on his own initiative, ~~~~ *invite* Japan to immediate adherence and at the same time mediate between Japan and ourselves. Please call the President's attention to the fact that the ruthless employment of our submarines now offers the prospect of compelling England in a few months to make peace." Signed, ZIMMERMANN.

Security Models: types of cipher

- Random functions – hash functions

  : accepts an input string of any length and outputs a string of fixed length, say *n* bits long

- Random generators – stream ciphers

  : (reverse of hash function?) accepts a short input and produces a long output

- Random permutations – block ciphers

  : keyed-invertible function transfers a fixed-size input to a fixed size output and vice-versa

- Public key encryption(special kind of block cipher):

  : encrypt for anyone, but decrypt for only key owner

- Digital signatures

  : sign by only key owner, but checked by anyone

Cryptographic Hash Function

- Cryptographic hash function $h(x)$ must provide
  - **Compression** — output length is small
  - **Efficiency** — easy to compute $h(x)$ for any x
  - **One-way** — given a value y it is infeasible to find an x such that $h(x) = y$
  - **Weak collision resistance** — given x and $h(x)$, infeasible to find $y \neq x$ such that $h(y) = h(x)$
  - **Strong collision resistance** — infeasible to find ***any*** x and y, with $x \neq y$ such that $h(x) = h(y)$
- Lots of collisions exist, but hard to find ***any***
- https://www.convertstring.com/ko/Hash/SHA256

Pre-Birthday Problem

- Suppose N people in a room
- How large must N be before the probability someone has same birthday as me is $\geq 1/2$?
  - Solve: $1/2 = 1 - (364/365)^N$ for N
  - We find $N = 253$

Birthday Paradox

With 23 random people in a room, there is a 0.507 chance that two share a birthday. This perhaps surprising observation has important implications for the design of cryptographic systems.

If we randomly throw k balls into n bins, then the probability that no bin contains two balls is

$$\left(1-\frac{1}{n}\right)\cdot\left(1-\frac{2}{n}\right)\cdots\left(1-\frac{k-1}{n}\right)=\prod_{i=1}^{k-1}\left(1-\frac{i}{n}\right)=\frac{n!}{(n-k)!\cdot n^k}$$

It can be shown that this probability is less than $\frac{1}{2}$ if k is slightly above $\sqrt{n}$. As $n \to \infty$, the expected number of balls needed for a collision is $\sqrt{n\pi/2}$.

One consequence is that if a 2^n search is considered computationally sufficiently infeasible, then the output of a collision-resistant hash function needs to be at least $2n$ bits large.

from Introduction to Security by Markus Kuhn
http://www.cl.cam.ac.uk/teaching/0708/IntroSec/

Of Hashes and Birthdays

- If $h(x)$ is N bits long, 2^N different hash values are possible
- So, if you hash about $2^{N/2}$ random values then you expect to find a collision
  - Since $\mathrm{sqrt}(2^N) = 2^{N/2}$
- **Implication:** secure N bit symmetric key requires 2^{N-1} work to "break" while secure N bit hash requires $2^{N/2}$ work to "break"
  - Using brute force attack(that is exhaustive key search)

Popular Cryptographic Hashes

- **MD5** — invented by Rivest (do NOT use!)
  - 128 bit output, collisions easy to find
- **SHA-1** — A U.S. government standard, inner workings similar to MD5 (broken in 2017, do NOT use!)
  - 160 bit output
- SHA-2 family
  - same basic algorithm with SHA-1(so?)
  - SHA-224, SHA-256, SHA-384, SHA-512
- SHA-3 family: *Keccak* as SHA-3 in 2015(NIST)
  - different math algorithm to SHA-1/2
  - SHA3-224, SHA3-256, SHA3-384, SHA3-512

Applications of Secure Hash Functions

Message Authentication Code

Hash a message M concatenated with a key K:

$$\mathrm{MAC}_K(M) = h(K, M)$$

A standard technique that is widely used with MD5 or SHA-1 for h is:

$$\mathrm{HMAC}_K = h(K \oplus X_1, h(K \oplus X_2, M))$$

The fixed padding values X_1, X_2 used in HMAC extend the length of the key to the input size of the compression function, thereby permitting precomputation of its first iteration.

`http://www.ietf.org/rfc/rfc2104.txt`

from Introduction to Security by Markus Kuhn
http://www.cl.cam.ac.uk/teaching/0708/IntroSec/

Hash Usage Example: Spam Reduction

- Spam reduction
- Before accept email, want proof that sender spent effort to create email
  - Here, effort == CPU cycles
- Goal is to limit the amount of email that can be sent
  - This approach will not eliminate spam
  - Instead, make spam more costly to send

Hash Usage Example: Spam Reduction

- Let M = email message

  R = value to be determined

  T = current time
- Sender must find R so that

  $h(M,R,T) = (00\ldots0,X)$, where

  N initial bits of hash value are **all zero**
- Sender then sends (M,R,T)
- Recipient accepts email, provided that…

  $h(M,R,T)$ begins with N zeros

Hash Usage Example: Spam Reduction

- Sender: $h(M,R,T)$ begins with N zeros
- Recipient: verify that $h(M,R,T)$ begins with N zeros
- **Work for sender:** about 2^N **hashes**
- **Work for recipient:** always 1 **hash**
- Sender's work increases exponentially in N
- Small work for recipient regardless of N
- Choose N so that…
  - Work acceptable for normal email users
  - Work is too high for spammers

Q & A

aiclasshongik@gmail.com