IT-52042

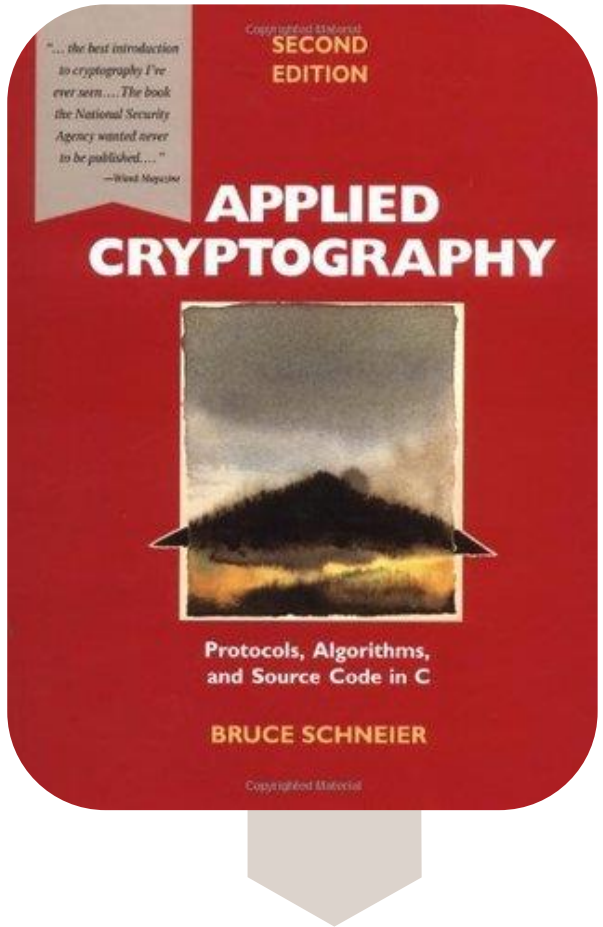
Cryptography and Network Security

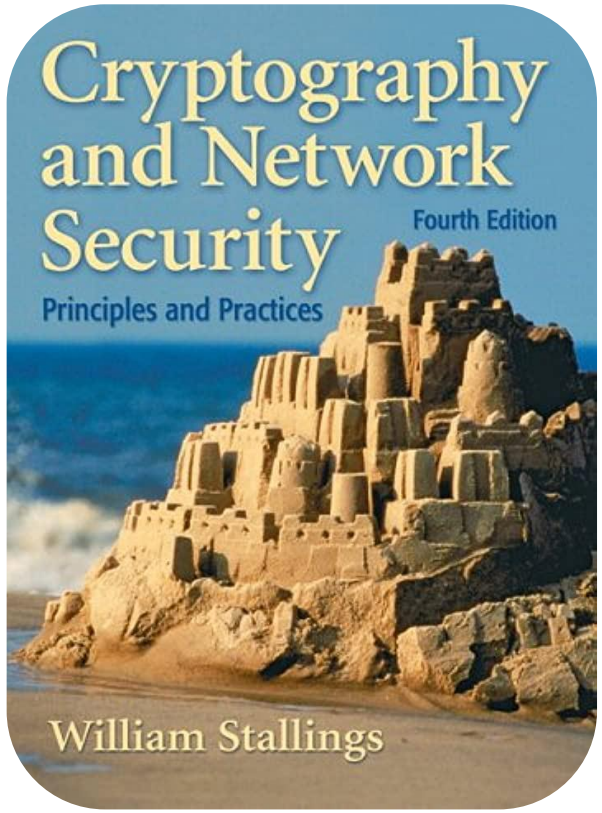
Daw Win Pa Pa Phyo

Demonstrator

Department of Information Technology Engineering

West Yangon Technological University





Chapter-7: Key Length

Chapter Outlines:

- What is key length?
- Symmetric Key Length
- Public-Key Length
- Comparing Symmetric and Public-Key Length
- Birthday Attacks against One-way hash Function
- How long should a Key be?

Objectives

- To understand the important of key length for the security of a cryptosystem
- To understand about symmetric key length and public-key length
- To comprehend various attacks trying every possible key to break the cryptosystem
- To study the facts how long a key should be

What is key length?

- Key length is an essential aspect of cryptography, and it determines the security strength of an encryption algorithm. It is typically measured **in bits. (2^n)**
- The length of the key directly affects the security of the encryption.
- Various cryptographic standards and organizations, such as NIST (National Institute of Standards and Technology), publish guidelines and recommendations for key lengths based on their evaluation of current computing capabilities and security requirements.
- In symmetric encryption algorithms like AES (Advanced Encryption Standard), common key lengths include 128 bits, 192 bits, and 256 bits. In asymmetric encryption algorithms like RSA or ECC (Elliptic Curve Cryptography), key lengths of 2048 bits, 3072 bits, or even higher are commonly used.

Symmetric Key Length

❖ The security of a symmetric cryptosystem is a function of two things: **the strength of the algorithm and the length of the key.**

❖ Symmetric key length in cryptography refers to the size or length of the secret key used in symmetric encryption algorithms.

❖ The recommended key lengths for symmetric encryption algorithms depend on several factors, including the algorithm itself, the desired level of security, and advancements in computing power.

❖ Longer key lengths offer increased security but may require more computational resources and potentially impact the system's performance. It is important to strike a balance between **security requirements and operational efficiency** when selecting a symmetric key length for a given application.

Symmetric Key Length

❖ **Advanced Encryption Standard (AES):** AES is one of the most widely used symmetric encryption algorithms. It supports key lengths of **128 bits, 192 bits, and 256 bits**. AES-128 is considered secure for most applications, while AES-256 offers a higher level of security.

❖ **Triple Data Encryption Standard (3DES):** 3DES is a symmetric encryption algorithm that applies the Data Encryption Standard (DES) algorithm multiple times with different keys. It typically uses a key length of **168 bits (three 56-bit keys).** However, due to its relatively shorter key length, 3DES is considered less secure than AES.

❖ **Blowfish**: Blowfish is a symmetric encryption algorithm that supports variable key lengths, **ranging from 32 bits to 448 bits**. It is known for its flexibility in key length and is considered secure when using key lengths of 128 bits or higher.

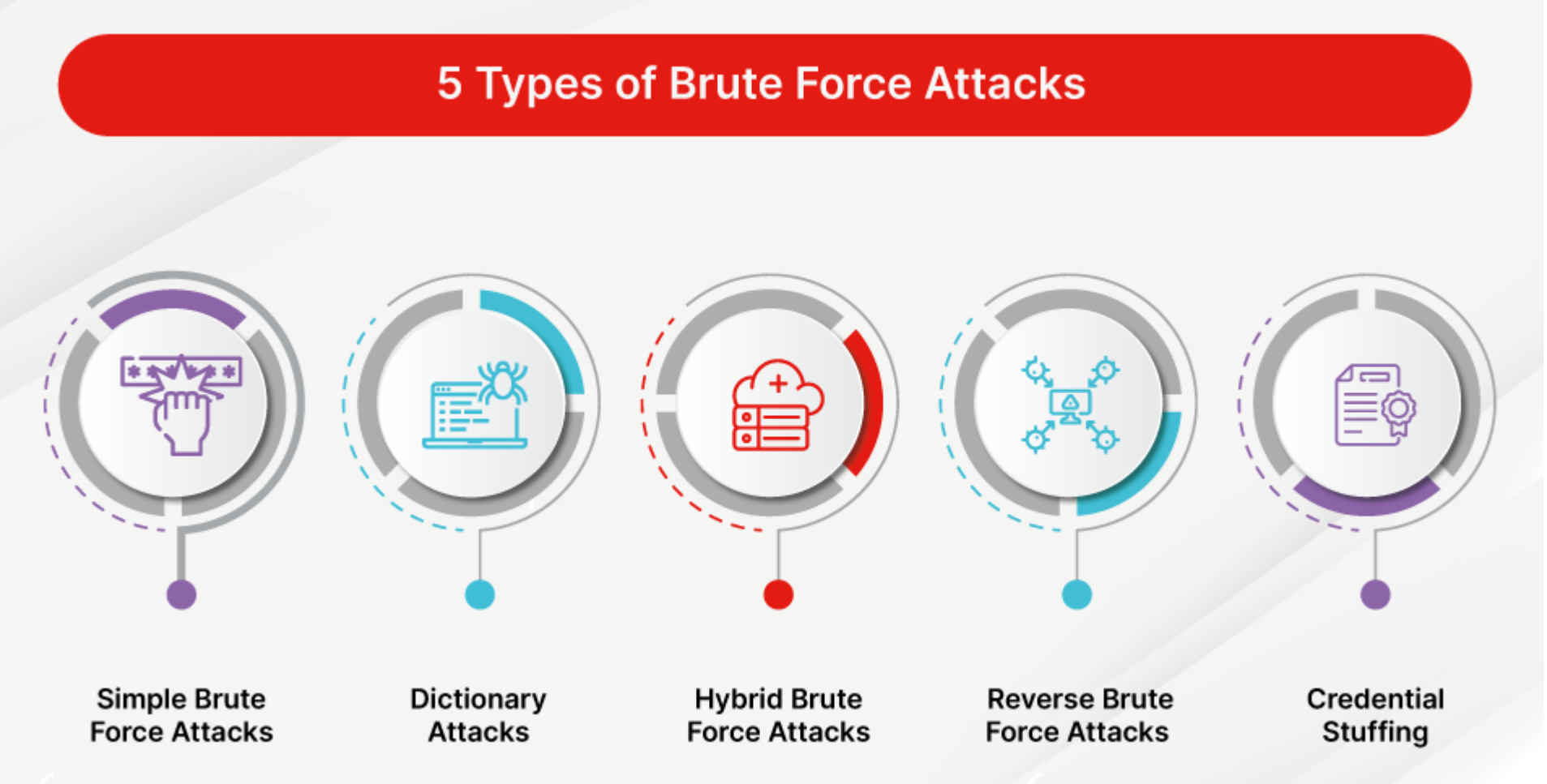
Bute-Force Attack

❖ A brute-force attack is a trial-and-error method used by application programs to decode login information and encryption keys to use them to gain unauthorized access to systems.

❖ The name "brute force" comes from attackers using **excessively forceful attempts** to gain access to user accounts. Despite being an old cyberattack method, brute force attacks are tried and tested and remain a popular tactic with hackers.

Bute-Force Attack (Cont'd)

Types of Brute Force Attacks



Bute-Force Attack (Cont'd)

Why do Brute Force attack occur?

❖ Hackers want to get into other people's systems for many reasons. Although sometimes their intentions can be unknown or personal, from general assumptions, here are a few common reasons why a brute force attack occurs.

- ❑ Exploit Activity Data for Financial Gains
- ❑ Gain Access to Personal Data
- ❑ Spreading Malware
- ❑ Damage a Company's Reputation

Bute-Force Attack (Cont'd)

How to prevent Bruce Force Attack?

Bute-Force Attack (Cont'd)

5 Best Brute Force Attack Tools for Penetration Testing

- BruteX
- Disreach
- Callow
- SSB
- Brup Suite Professional

Bute-Force Attack

- Cryptanalyst needs a small amount of ciphertext and the corresponding plaintext;
- a brute-force attack is **a known-plaintext attack**.
- For a block cipher, the cryptanalyst would need a block of ciphertext and corresponding plaintext: generally 64 bits.

Bute-Force Attack

2 bits long key, 2^2
00
01
10
11

Calculating the complexity of brute force attack:

- If the key is 8 bits long, there are 28, or 256, possible keys. It will take 256 attempts to find the correct key,
- If the key is 56 bits long, then there are 2^{56} possible keys. Assuming a supercomputer can try a million keys a second, it will take 2285 years to find the correct key
- If the key is 64 bits long, then it will take the same supercomputer about 585,000 years to find the correct key among the 264 possible keys.
- If the key is 128 bits long, it will take 10^{25} years.

Bute-Force Attack

Consideration on security of cryptosystems

- Cryptosystems that look perfect are often extremely weak.
- Strong cryptosystems, with a couple of minor changes, can become weak.
- It is best to trust algorithms that professional cryptographers have scrutinized for years without cracking them and to be suspicious of algorithm designers' grandiose claims of security.

Time and Cost Estimates for Brute-Force Attack

- Two parameters determine the speed of a brute-force attack: **the number of keys to be tested** and **the speed of each test.**
- Most symmetric algorithms accept any fixed-length bit pattern as the key.
- The speed at which each possible key can be tested is also a factor.
- Different algorithm can be tested in the same amount of time.
- Since we are looking for key lengths that are millions of times more difficult to crack than would be feasible, small differences due to test speed are irrelevant.

Time and Cost Estimates for Brute-Force Attack (cont'd)

- **Special-purpose DES-cracking machine**
  - This machine could test 2^{56} keys in 20 hours and if built to attack an algorithm with a 64-bit key, it could test all 2^{64} keys in 214 days.
- **Parallel processors**
  - Each processor can test a subset of the keyspace.
  - The processors do not have to communicate among themselves; the only communication required at all is a single message signifying success.
  - There are no shared memory requirements.
  - It is easy to design a machine with a million parallel processors, each working independent of the others.

Time and Cost Estimates for Brute-Force Attack (cont'd)

Average Time Estimates for a Hardware Brute-Force Attack in 1995

Cost	Length of Key in Bits: 40	56	64	80	112	128
$100 K	2 seconds	35 hours	1 year	70,000 years	10^{14} years	10^{19} years
$1 M	.2 seconds	3.5 hours	37 days	7000 years	10^{13} years	10^{18} years
$10 M	.02 seconds	21 minutes	4 days	700 years	10^{12} years	10^{17} years
$100 M	2 milliseconds	2 minutes	9 hours	70 years	10^{11} years	10^{16} years
$1 G	.2 milliseconds	13 seconds	1 hour	7 years	10^{10} years	10^{15} years
$10 G	.02 milliseconds	1 second	5.4 minutes	245 days	10^{9} years	10^{14} years
$100 G	2 microseconds	.1 second	32 seconds	24 days	10^{8} years	10^{13} years
$1 T	.2 microseconds	.01 second	3 seconds	2.4 days	10^{7} years	10^{12} years
$10 T	.02 microseconds	1 millisecond	.3 second	6 hours	10^{6} years	10^{11} years

Software Crackers

- A software cracker **(a software cracking tool or a reverse engineer)** is an individual or a piece of software that aims to bypass or remove the copy protection measures or licensing restrictions imposed on software. While software cracking is not directly related to cryptography, it can sometimes involve techniques that touch upon cryptographic aspects.
- Without special-purpose hardware and massively parallel machines, brute-force attacks are significantly harder.
- A software attack is about a thousand times slower than a hardware attack.
- It costs nothing to set up a microcomputer to test possible keys whenever it is idle.
- If it finds the correct key—great. If it doesn't, then nothing is lost. It

Software Crackers (cont'd)

- Imagine a university computer network of 512 workstations (a medium-sized network)
- They could even be spread around the world, coordinating their activity through electronic mail.
- Assume each workstation is capable of running [the algorithm] at a rate of 15,000 encryptions per second.
- Allowing for the overhead of testing and changing keys, this comes down to...8192 tests per second per machine.
- To exhaust [a 56-bit] keyspace with this setup would take 545 years (assuming the network was dedicated to the task twenty-four hours per day).

Software Crackers (cont'd)

- The same calculations give our hypothetical student hackers one chance in 200,000 of cracking a key in one day.
- Over a long weekend their odds increase to one chance in sixty-six thousand.
- The faster their hardware, or the more machines involved, the better their chance becomes.
- Using an algorithm with a 64-bit key instead of a 56-bit key makes this attack 256 times more difficult.
- A 128-bit key makes a brute-force attack ridiculous even to contemplate

Neural Networks

- Neural nets aren't terribly useful for cryptanalysis, primarily because of the shape of the solution space.
- This allows a neural net to learn, proposing better and better solutions as it does.
- Breaking an algorithm provides for very little in the way of learning opportunities:

Viruses

- The greatest difficulty in getting millions of computers to work on a brute-force attack is convincing millions of computer owners to participate.
- You could also use a computer virus to spread the cracking program more efficiently over as many computers as possible.
- The attacker writes and lets loose a computer virus.
- This virus doesn't reformat the hard drive or delete files; it works on a brute-force cryptanalysis problem whenever the computer is idle.
- Microcomputers are idle between 70 percent and 90 percent of the time.

Viruses (cont'd)

- Eventually, one machine will stumble on the correct key.
- Two ways of proceeding:
  - **First**, the virus could spawn a different virus. It wouldn't do anything but reproduce and delete any copies of the cracking virus it finds but would contain the information about the correct key. This new virus would simply propagate through the computer world until it lands on the computer of the person who wrote the original virus.
  - A second, sneakier approach would be for the virus to display this message on the screen.

Viruses (cont'd)

- How efficient is this attack?
  - Assume the typical infected computer tries a thousand keys per second.
  - The typical virus infects 10 million machines.
  - This virus can break a 56-bit key in 83 days and a 64-bit key in 58 years.

The Chinese Lottery

- The Chinese Lottery is an eclectic.
- A brute-force, million-test-per-second cracking chip was built into every radio and television sold.
- Each chip is programmed to test a different set of keys automatically upon receiving a plaintext/ciphertext pair over the airwaves.
- If every man, woman, and child in China owns a radio or television, then the correct key to a 56-bit algorithm will appear in 61 seconds.
- If only 1 in 10 Chinese owns a radio or television—closer to reality—the correct key will appear in 10 minutes.

The Chinese Lottery (cont'd)

- Some modifications are required to make this attack practical.
- To have each chip try random keys instead of a unique set of keys.
- This would make the attack about 39 percent slower.

Table 7.2
Brute-Force Cracking Estimates for Chinese Lottery

Country	Population	# of Televisions/Radios	Time to Break	
			56-bit	64-bit
China	1,190,431,000	257,000,000	280 seconds	20 hours
U.S.	260,714,000	739,000,000	97 seconds	6.9 hours
Iraq	19,890,000	4,730,000	4.2 hours	44 days
Israel	5,051,000	3,640,000	5.5 hours	58 days
Wyoming	470,000	1,330,000	15 hours	160 days
Winnemucca, NV	6,100	17,300	48 days	34 years

7.2 Public-Key Key Length

- Public-key key length refers to the size or length of the cryptographic keys used in public-key cryptography.
- The security strength of public-key cryptography relies on the computational difficulty of certain mathematical problems, such as factoring large numbers or the discrete logarithm problem.
- Common key lengths for RSA include 2048 bits, 3072 bits, and 4096 bits. **RSA-2048** is widely used for many applications, while RSA-3072 and RSA-4096 offer higher security levels.
- Common ECC key lengths include 256 bits, 384 bits, and 521 bits.

7.2 Public-Key Key Length

- Multiplying two large primes is a one-way function; it's easy to multiply the numbers to get a product but hard to factor the product and recover the two large primes.
- Public-key cryptography uses this idea to make a trap-door one-way function.
- Today's dominant public-key encryption algorithms are based on the difficulty of factoring large numbers that are the product of two large primes.
- Breaking these algorithms does not involve trying every possible key; breaking these algorithms involves trying to factor the large number (or taking discrete logarithms in a very large finite field—a similar problem). If the number is too small, you have no security. If the number is large enough, you have security against all the computing power in the world working from now until the sun goes nova.

7.2 Public-Key Key Length
Factoring Problem

- Factoring large number is hard.
- The fastest factoring algorithm during the time was the **quaratic sleve**.

Table 7.3
Factoring Using the Quadratic Sieve

Year	# of decimal digits factored	How many times harder to factor a 512-bit number
1983	71	>20 million
1985	80	>2 million
1988	90	250,000
1989	100	30,000
1993	120	500
1994	129	100

7.2 Public-Key Key Length

Factoring Problem

- Computing power is generally measured in **mips-years:** a one-million-instruction-per-second (mips) computer running for one year, or about 3*1013 instructions.
- A 1-mips machine is equivalent to the DEC VAX 11/780. Hence, a mips-year is a VAX 11/780 running for a year, or the equivalent. (A 100 MHz Pentium is about a 50 mips machine; a 1800-node Intel Paragon is about 50,000.)
- The 1983 factorization of a 71-digit number required 0.1 mips-years; the 1994 factorization of a 129- digit number required 5000.
- A new factoring algorithm has taken over from the quadratic sieve**: the general number field sieve.**
- The general number field sieve can factor a 512-bit number over 10 times faster than the quadratic sieve.

7.2 Public-Key Key Length Factoring Problem

Table 7.4
Factoring Using the General Number Field Sieve

# of bits	Mips-years required to factor
512	30,000
768	$2*10^8$
1024	$3*10^{11}$
1280	$1*10^{14}$
1536	$3*10^{16}$
2048	$3*10^{20}$

Table 7.5
Factoring Using the Special Number Field Sieve

# of bits	Mips-years required to factor
512	<200
768	100,000
1024	$3*10^7$
1280	$3*10^9$
1536	$2*10^{11}$
2048	$4*10^{14}$

Why not use 10,000-bit key? 

Remember that you pay a price in computation time as your keys get longer and enough to be computationally useable.

7.2 Public-Key Key Length
Factoring Problem

Table 7.6
Recommended Public-key Key Lengths (in bits)

Year	vs. Individual	vs. Corporation	vs. Government
1995	768	1280	1536
2000	1024	1280	1536
2005	1280	1536	2048
2010	1280	1536	2048
2015	1536	2048	2048

Table 7.7
Long-range Factoring Predictions

Year	Key Length (in bits)
1995	1024
2005	2048
2015	4096
2025	8192
2035	16,384
2045	32,768

7.3 Comparing Symmetric and Public-Key Key Length

- If you are designing a system that uses both symmetric and public-key cryptography, the key lengths for each type of cryptography should be chosen so that it is equally difficult to attack the system via each mechanism.

Table 7.9
Symmetric and Public-key Key Lengths with Similar Resistances to Brute-Force Attacks

Symmetric Key Length	Public-key Key Length
56 bits	384 bits
64 bits	512 bits
80 bits	768 bits
112 bits	1792 bits
128 bits	2304 bits

7.4 Birthday Attacks against One-Way hash Functions

- Finding two people with the same random birthday is analogous to the **birthday attack.**
- Assume that a one-way hash function is secure and the best way to attack it by using brute force.
- It produces an **m-bit output.**
- Finding a message that hashes to a given hash value would require hashing 2m random messages. Finding two messages that hash to the same value would only require hashing 2m/2 random messages. A machine that hashes a million messages per second would take 600,000 years to find a second message that matched a given 64-bit hash. The

Brute-force attacks against a one-way hash function

- There are two brute-force attacks against a one-way hash function.
- The first is the most obvious:
  - Given the hash of message, H(M), an adversary would like to be able to create another document, M′, such that H(M) = H(M′). The second attack is more subtle: An adversary would like to find two random messages, M, and M′, such that H(M) = H(M′).
- The second attack is:
  - An adversary would like to find two random messages, M, and M′,
  - H(M) = H(M′). This is called a **collision,**

7.5 How long should a key be?

- It depends on the situation.
- To determine,
  - How much security you need
  - How much is your data worth?
  - How long does it need to be secure?
  - What are your adversaries' resources?

Table 7.10
Security Requirements for Different Information

Type of Traffic	Lifetime	Minimum Key Length
Tactical military information	minutes/hours	56–64 bits
Product announcements, mergers, interest rates	days/weeks	64 bits
Long-term business plans	years	64 bits
Trade secrets (e.g., recipe for Coca-Cola)	decades	112 bits
H-bomb secrets	>40 years	128 bits
Identities of spies	>50 years	128 bits
Personal affairs	>50 years	128 bits
Diplomatic embarrassments	>65 years	at least 128 bits
U.S. census data	100 years	at least 128 bits

References

- https://www.tutorialspoint.com/cryptography/
- https://www.youtube.com/@sunnylearning
- https://www.cloudways.com/blog/what-is-brute-force-attack/
- Applied Cryptography (2nd Edition)
- Refresher Course of Cryptography & Network Security

Thank you

dawwinpapaphyo22@gmail.com