CHAPTER 7

Key Length

Symmetric Key Length

- The security of a symmetric cryptosystem is a function of two things:
  1. The strength of the algorithm. (it is more important) and
  2. The length of the key.
- If the strength of the algorithm is perfect, then there is no better way to break the cryptosystem other than trying every possible key in a brute-force attack.

- Brute-force attack is a known plaintext attack (i.e. ciphertext and the corresponding plaintext is required).
- The cryptanalyst does not need much plaintext to launch this attack.

Symmetric Key Length Continue…

- Calculating the complexity of a brute-force attack is easy.
- For key length 8 bits, there are 2^8=256 possible keys.
- For key length 56 bits, there are 2^{56} possible keys (then for a supercomputer with a speed of million keys/s need 2285 years).

- For key length 64 bits, the supercomputer needs 585,000 years to find the correct key among 2^{64} possible keys.
- For 128 bits, it takes 10^{25} years.
- For 1024 bits, a million million-attempts-per-second computers working in parallel will need 10^{597} years.

Time and Cost estimates of Brute-force attack

- Brute-force attack is a known plaintext attack.
- The speed depends on the following two parameter:
  1. The no. of keys to be tested. And
  2. The speed of each test (less important).

- According to Moore's law: Computing power doubles approximately every 18 months. This means cost go down a factor of 10 every five years; what cost $1 million to build in 2010 will cost a mere $1 lac in the year 2015.

- A table is shown on the next slide about average time estimates for a Hardware brute-force attacks in 1995.

Time and Cost estimates of Brute-force attack Continue…

Average Time Estimates for a Hardware Brute-force Attacks in1995

Length of Key in Bits

Cost	40	56	64	80	112	128
$100K	2s	35h	1year	70,000y	10^{14}year	10^{19}year
$1M	0.2s	3.5h	37day	7,000y	10^{13}year	10^{18}year
$10M	0.02s	21m	4day	700y	10^{12}year	10^{17}year
$100M	2ms	2m	9h	70y	10^{11}year	10^{16}year
$1G	0.2ms	13s	1h	7y	10^{10}year	10^{15}year
$10G	0.02ms	1s	5.4m	245day	10^{9}year	10^{14}year
$100G	2µs	0.1s	32s	24day	10^{8}year	10^{13}year
$1T	0.2µs	0.01s	3s	2.4day	10^{7}year	10^{12}year
$10T	0.02µs	1ms	0.3s	6	10^{6}year	10^{11}year

Software Crackers

- 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.

Public-key key Length

- Public key 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; rather they involves trying to factor the large numbers.

- If the number is too small, you have no security.
- In 1977 Ron Rivest said that factoring a 125 digit number would take 40 quadrillion years.
- In 1994 a 129 digit number was factored.

Public-key key Length Continue...

- Table below shows factoring records over the past dozen years.

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	2,50,000
1989	100	30,000
1993	120	500
1994	129	100

Birthday Attacks Against One-way Hash Functions

- There are two brute-force attacks against a one way hash functions.
  1. Given the hash of message, H(M), an adversaries would like to be able to create another document, M', such that H(M)=H(M').
  2. An adversaries would like to find two random messages, M, and M', such that H(M)=H(M').
- The second version of attack is much easier.

- The birthday paradox is a standard statistics problem. How many people must be in a room for the chance to be greater than even that one of them shares your birthday? The answer is 253, now how many people must be there be for the chance to be greater than even that at least two of them will share the same birthday? The answer is 23. With only 23 people in a room, there are still 253 different pairs of people in the room.

Birthday Attacks Against One-way Hash Functions Continue…

- Finding someone with a specific birthday is analogous to the first attack; finding two people with the same random birthday is analogous to the second attack.
- The second attack is known as birthday attack.
- Consider a one-way hash function that produce an m-bit output. Finding a message that hashes to a given hash value would require hashing 2^m random messages. Finding two messages that hash to the same value would only require hashing $2^{m/2}$ random messages.
- A machine with million messages/s would take 600,000 years to find a second message that matched a given 64-bit hash.
- The same machine could find a pair of messages that hashed to the same value in about an hour.