Chapter 11

Message Integrity and Message Authentication

11-1 MESSAGE INTEGRITY

The cryptography systems that we have studied so far provide secrecy, or confidentiality, but not integrity. However, there are occasions where we may not even need secrecy but instead must have integrity.

Topics discussed in this section:

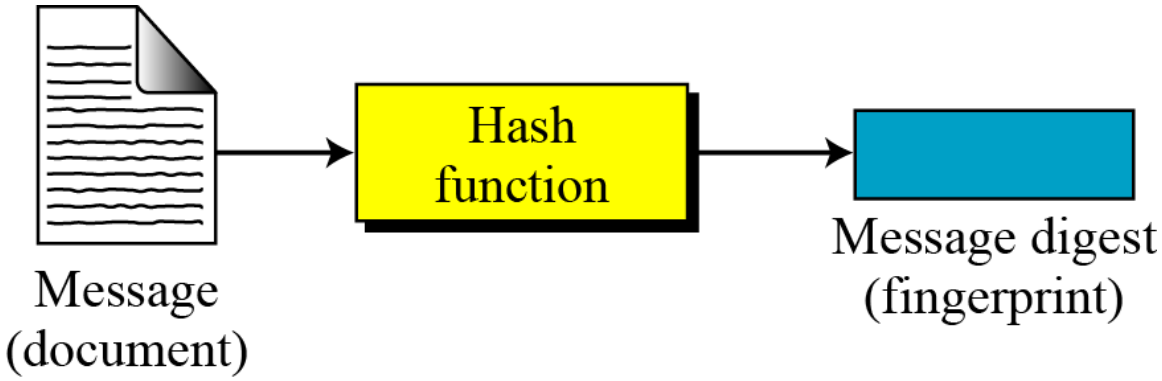
11.1.1 Document and Fingerprint

One way to preserve the integrity of a document is through the use of a fingerprint. If Alice needs to be sure that the contents of her document will not be changed, she can put her fingerprint at the bottom of the document.

11.1.2 Message and Message Digest

The electronic equivalent of the document and fingerprint pair is the message and digest pair.

Figure 11.1 *Message and digest*



11.1.3 Difference

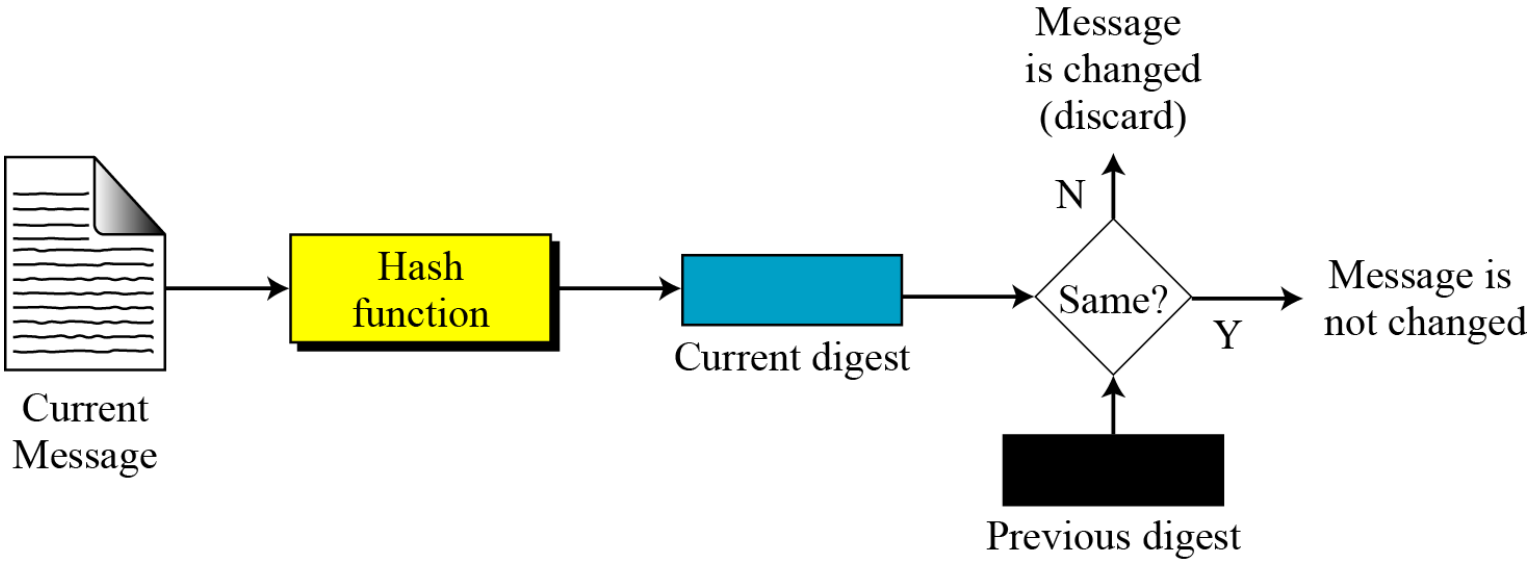
The two pairs (document / fingerprint) and (message / message digest) are similar, with some differences. The document and fingerprint are physically linked together. The message and message digest can be unlinked separately, and, most importantly, the message digest needs to be safe from change.

Note

The message digest needs to be safe from change.

11.1.4 Checking Integrity

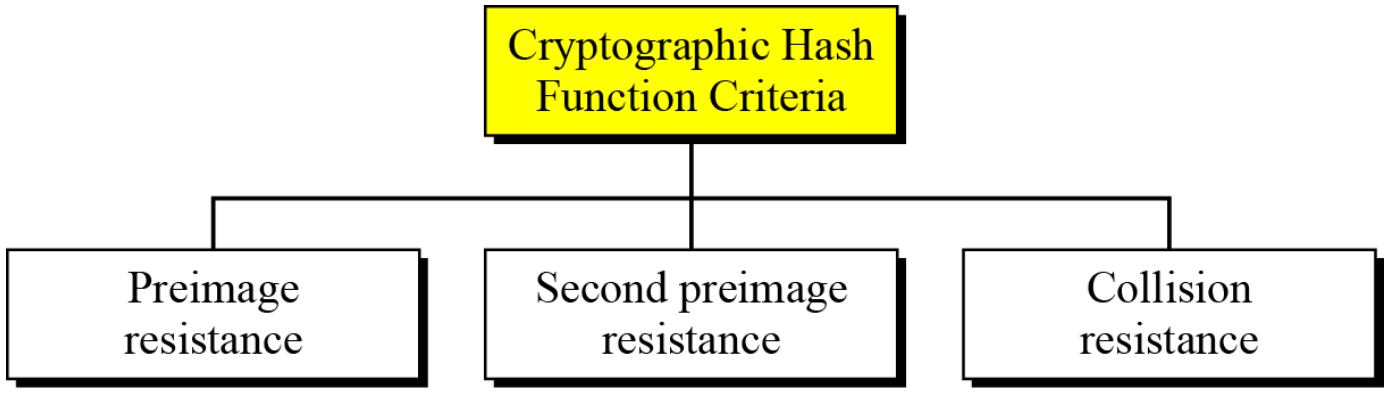
Figure 11.2 *Checking integrity*



11.1.5 Cryptographic Hash Function Criteria

A cryptographic hash function must satisfy three criteria: preimage resistance, second preimage resistance, and collision resistance.

Figure 11.3 *Criteria of a cryptographic hash function*



Preimage Resistance

Preimage Attack

Given: y = h(M) **Find: M′ such that y = h(M′)**

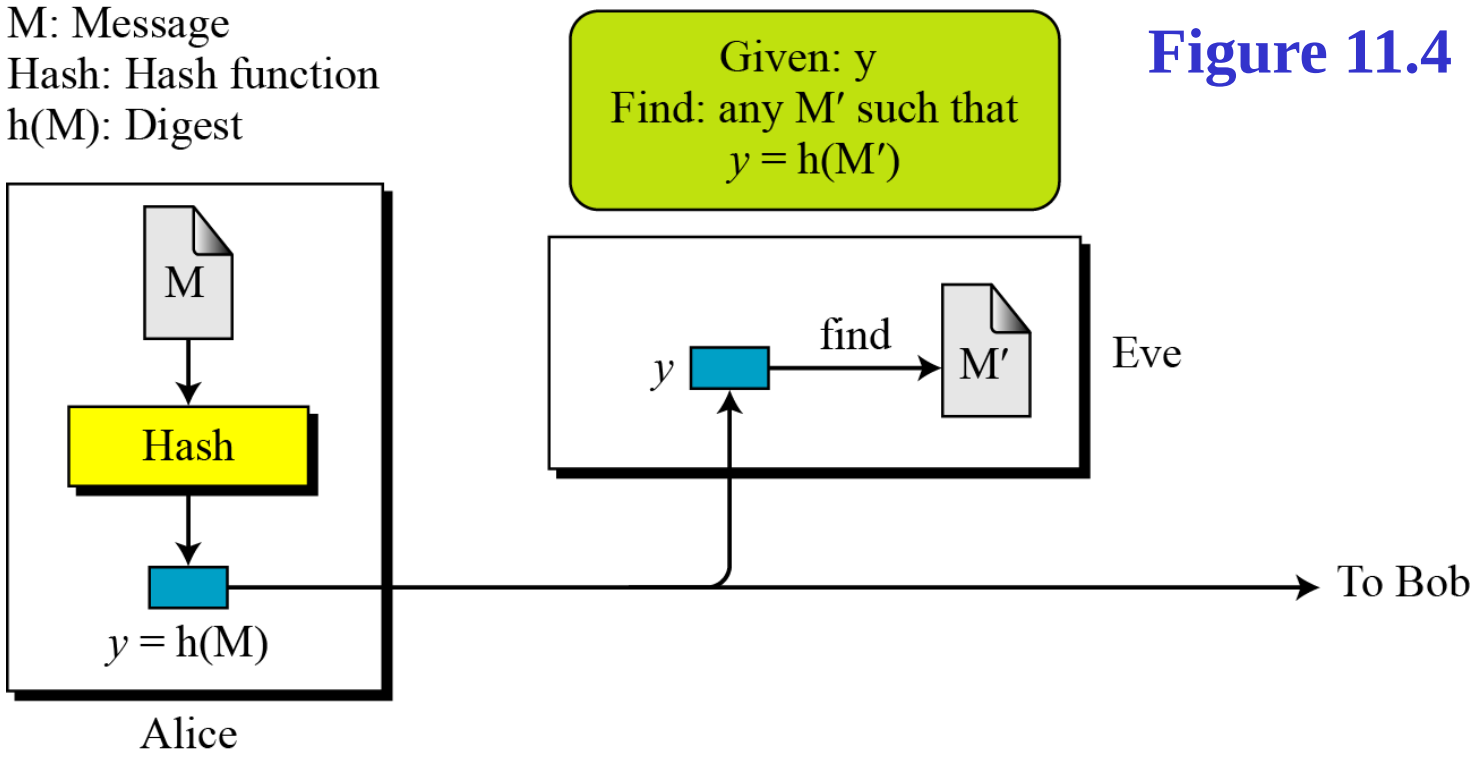


Figure 11.4 ***Preimage***

11.1.5 Continued

Example 11.1

Can we use a conventional lossless compression method as a cryptographic hash function?

Solution

We cannot. A lossless compression method creates a compressed message that is reversible.

Example 11.2

Can we use a checksum function as a cryptographic hash function?

Solution

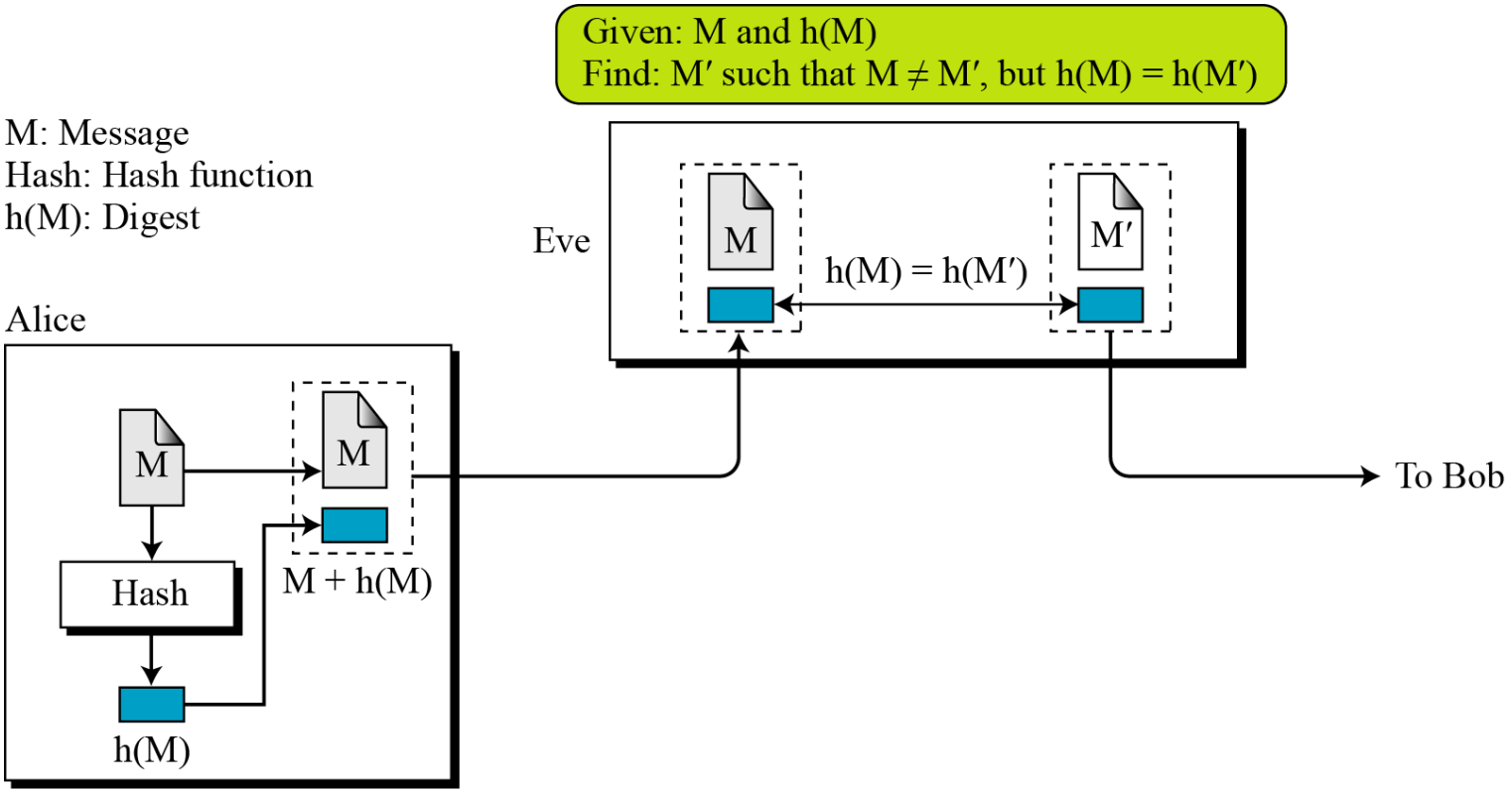
We cannot. A checksum function is not preimage resistant, Eve may find several messages whose checksum matches the given one.

Second Preimage Resistance

Second Preimage Attack

Given: M and h(M) **Find: M′ ≠ M such that h(M) = h(M′)**

Figure 11.5 *Second preimage*



Collision Resistance

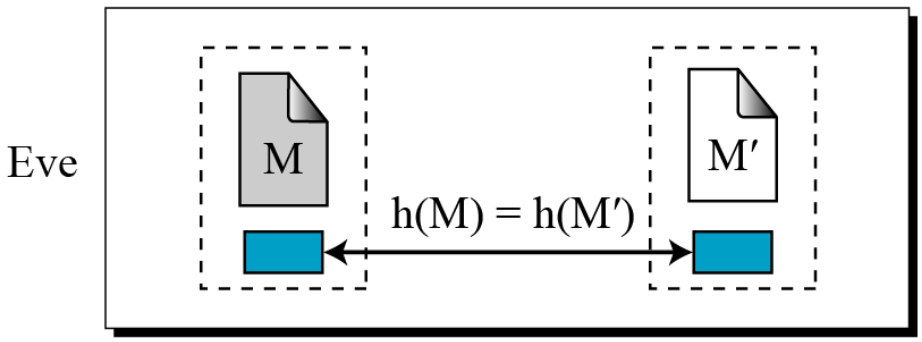
Collision Attack

Given: none **Find: $M' \neq M$ such that $h(M) = h(M')$**

Figure 11.6 *Collision*

M: Message
Hash: Hash function
h(M): Digest

Find: M and M′ such that $M \neq M'$, but $h(M) = h(M')$



11-2 RANDOM ORACLE MODEL

The Random Oracle Model, which was introduced in 1993 by Bellare and Rogaway, is an ideal mathematical model for a hash function.

Topics discussed in this section:

11.2.1 Pigeonhole Principle

If n pigeonholes are occupied by n + 1 pigeons, then at least one pigeonhole is occupied by two pigeons. The generalized version of the pigeonhole principle is that if n pigeonholes are occupied by kn + 1 pigeons, then at least one pigeonhole is occupied by k + 1 pigeons.

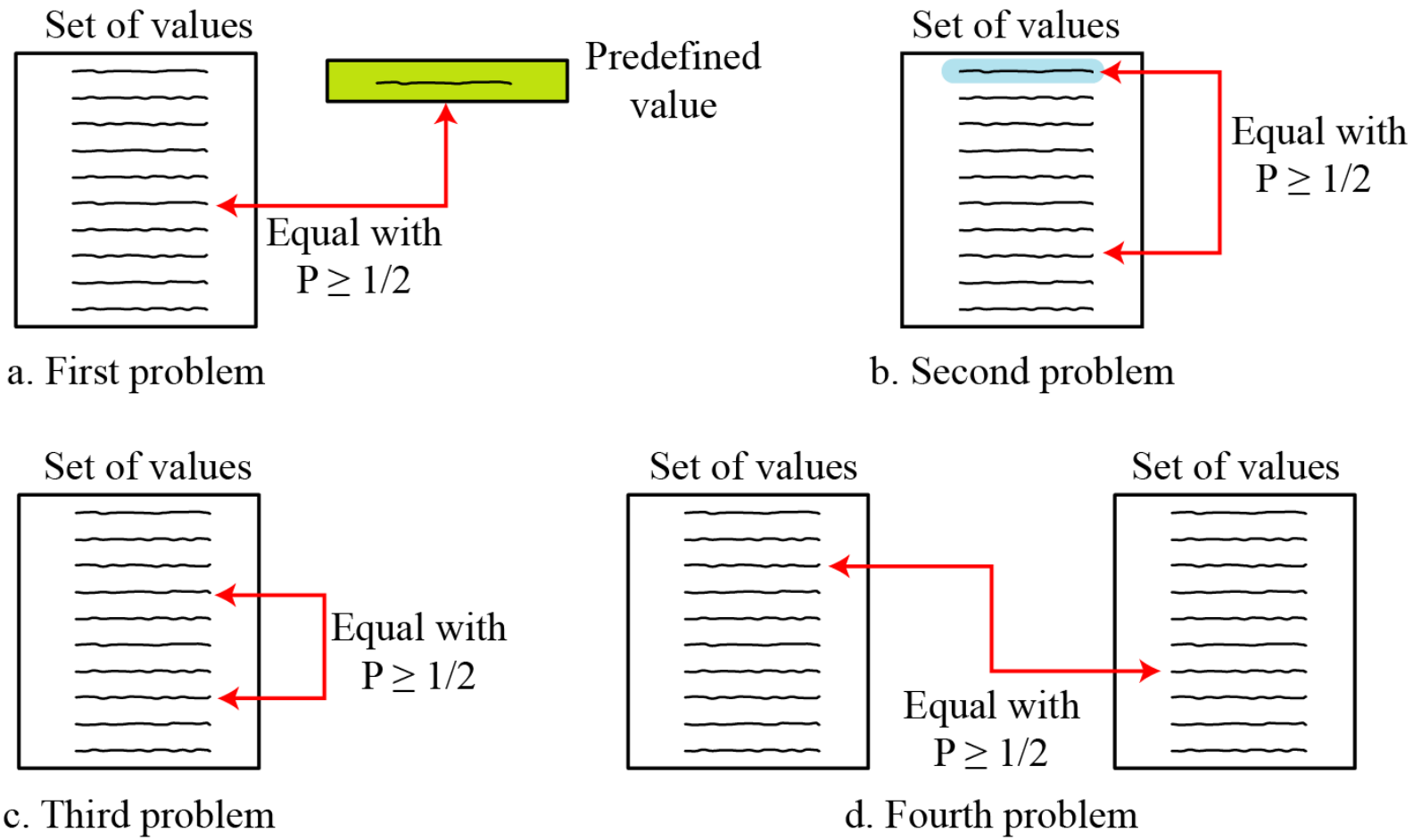
11.2.1 Continued

Example 11.5

Assume that the messages in a hash function are 6 bits long and the digests are only 4 bits long. Then the possible number of digests (pigeonholes) is $2^4 = 16$, and the possible number of messages (pigeons) is $2^6 = 64$. This means $n = 16$ and $kn + 1 = 64$, so k is larger than 3. The conclusion is that at least one digest corresponds to four ($k + 1$) messages.

11.2.2 Birthday Problems

Figure 11.7 *Four birthday problems*



11.2.2 Continued

Summary of Solutions

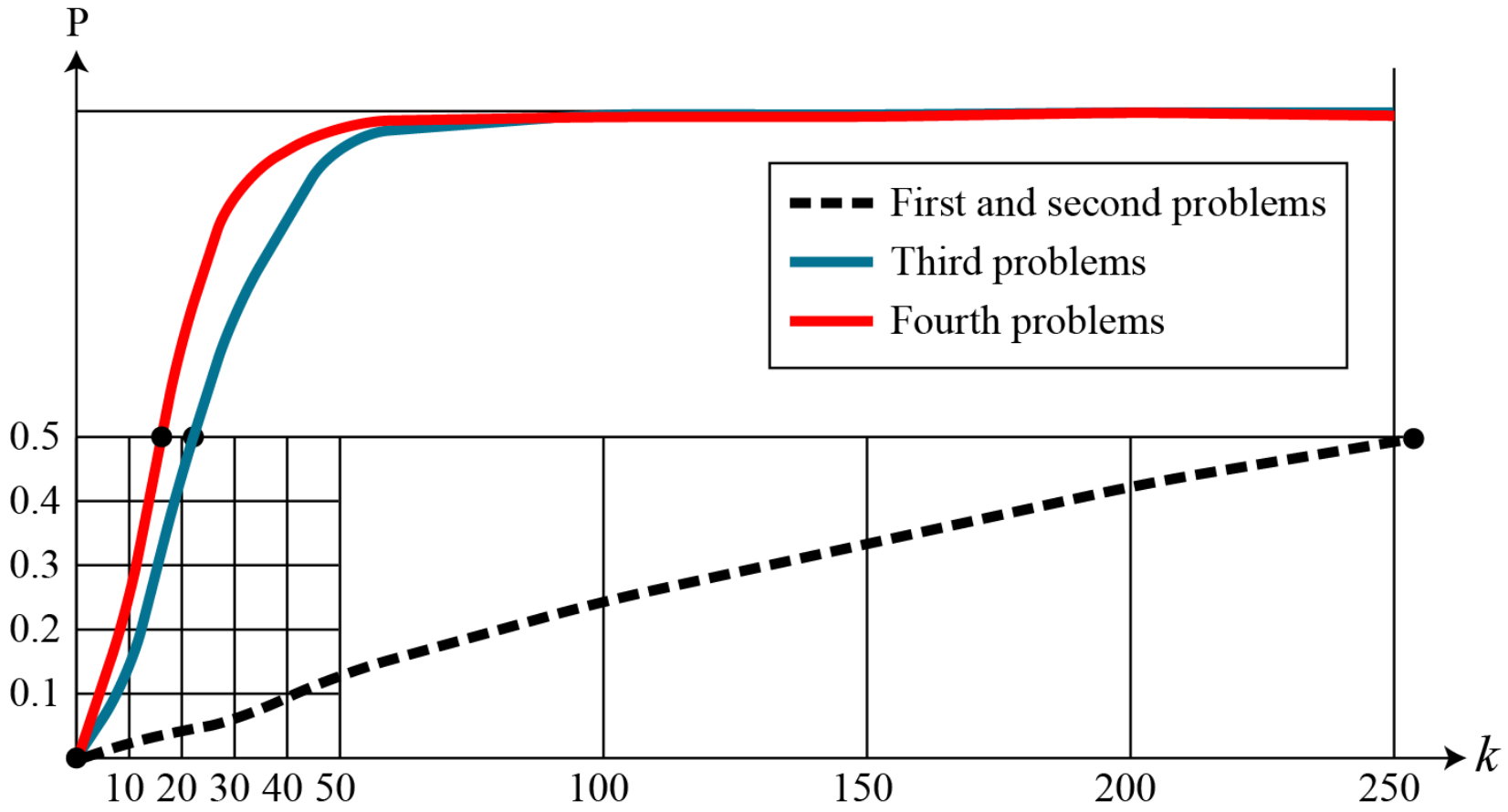
Solutions to these problems are given in Appendix E for interested readers; The results are summarized in Table 11.3.

Table 11.3 *Summarized solutions to four birthday problems*

Problem	*Probability*	*General value for k*	*Value of k with P = 1/2*	*Number of students (N = 365)*
1	$P \approx 1 - e^{-k/N}$	$k \approx \ln[1/(1-P)] \times N$	$k \approx 0.69 \times N$	253
2	$P \approx 1 - e^{-(k-1)/N}$	$k \approx \ln[1/(1-P)] \times N + 1$	$k \approx 0.69 \times N + 1$	254
3	$P \approx 1 - e^{-k(k-1)/2N}$	$k \approx \{2 \ln [1/(1-P)]\}^{1/2} \times N^{1/2}$	$k \approx 1.18 \times N^{1/2}$	23
4	$P \approx 1 - e^{-k^2/2N}$	$k \approx \{\ln [1/(1-P)]\}^{1/2} \times N^{1/2}$	$k \approx 0.83 \times N^{1/2}$	16

Comparison

Figure 11.8 *Graph of four birthday problem*



11.2.3 Attacks on Random Oracle Model

Preimage Attack

Algorithm 11.1 *Preimage attack*

```
Preimage_Attack (D)
{
    for (i = 1 to k)
    {
        create (M [i])
        T ← h(M [i])                    // T is a temporary digest
        if (T = D) return M [i]
    }
    return failure
}
```

The difficulty of a preimage attack is proportional to 2^n.

Example 11.6

A cryptographic hash function uses a digest of 64 bits. How many digests does Eve need to create to find the original message with the probability more than 0.5?

Solution

The number of digests to be created is $k \approx 0.69 \times 2^n \approx 0.69 \times 2^{64}$. This is a large number. Even if Eve can create 2^{30} (almost one billion) messages per second, it takes 0.69×2^{34} seconds or more than 500 years. This means that a message digest of size 64 bits is secure with respect to preimage attack, but, as we will see shortly, is not secured to collision attack.

Second Preimage Attack.

Algorithm 11.2 *Second preimage attack*

```
Second_Preimage_Attack (D, M)
{
    for (i = 1 to k – 1)
    {
        create (M [i])
        T ← h (M [i])                    // T is a temporary digest
        if (T = D) return M [i]
    }
    return failure
}
```

The difficulty of a second preimage attack is proportional to 2^n.

Collision Attack

Algorithm 11.3 *Collision attack*

```
Collision_Attack
{
   for (i = 1 to k )
   {
        create (M[i])
        D[i] ← h (M[i])                    // D [i] is a list of created digests
        for (j = 1 to i − 1)
        {
           if (D[i] = D[j]) return (M[i] and M[j])
        }
   }
   return failure
}
```

The difficulty of a collision attack is proportional to $2^{n/2}$.

11.2.3 Continued

Example 11.7

A cryptographic hash function uses a digest of 64 bits. How many digests does Eve need to create to find two messages with the same digest with the probability more than 0.5?

Solution

The number of digests to be created is $k \approx 1.18 \times 2^{n/2} \approx 1.18 \times 2^{32}$. If Eve can test 2^{20} (almost one million) messages per second, it takes 1.18×2^{12} seconds, or less than two hours. This means that a message digest of size 64 bits is not secure against the collision attack.

11.2.3 Continued

Summary of Attacks

Table 11.4 shows the level of difficulty for each attack if the digest is n bits.

Table 11.4 *Levels of difficulties for each type of attack*

Attack	*Value of k with P=1/2*	*Order*
Preimage	$k \approx 0.69 \times 2^n$	2^n
Second preimage	$k \approx 0.69 \times 2^n + 1$	2^n
Collision	$k \approx 1.18 \times 2^{n/2}$	$2^{n/2}$
Alternate collision	$k \approx 0.83 \times 2^{n/2}$	$2^{n/2}$

11.2.3 Continued

Example 11.9

MD5 (see Chapter 12), which was one of the standard hash functions for a long time, creates digests of 128 bits. To launch a collision attack, the adversary needs to test 2^{64} ($2^{128/2}$) tests in the collision algorithm. Even if the adversary can perform 2^{30} (more than one billion) tests in a second, it takes 2^{34} seconds (more than 500 years) to launch an attack.

It has been proved that MD5 can be attacked on less than 2^{64} tests because of the structure of the algorithm.

11.2.3 Continued

Example 11.10

SHA-1, a standard hash function developed by NIST, creates digests of 160 bits.

To launch a collision attack, the adversary needs to test $2^{160/2} = 2^{80}$ tests in the collision algorithm.

Even if the adversary can perform 2^{30} (more than one billion) tests in a second, it takes 2^{50} seconds (more than ten thousand years) to launch an attack.

11.2.3 Continued

Example 11.11

The new hash function, that is likely to become NIST standard, is SHA-512, which has a 512-bit digest. This function is definitely resistant to collision attacks based on the Random Oracle Model. It needs $2^{512/2} = 2^{256}$ tests to find a collision with the probability of 1/2.

11-3 MESSAGE AUTHENTICATION

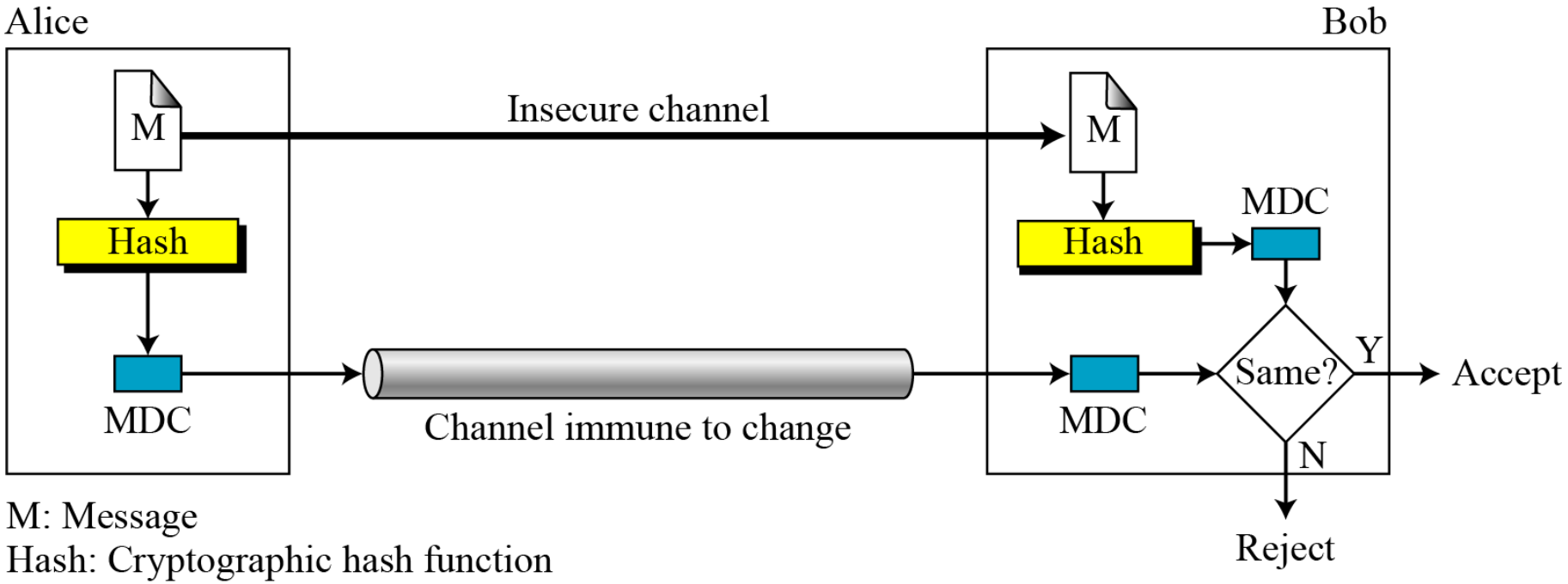
A message digest does not authenticate the sender of the message. To provide message authentication, Alice needs to provide proof that it is Alice sending the message and not an impostor. The digest created by a cryptographic hash function is normally called a modification detection code (MDC). What we need for message authentication is a message authentication code (MAC).

Topics discussed in this section:

11.3.1 Modification Detection Code (MDC)

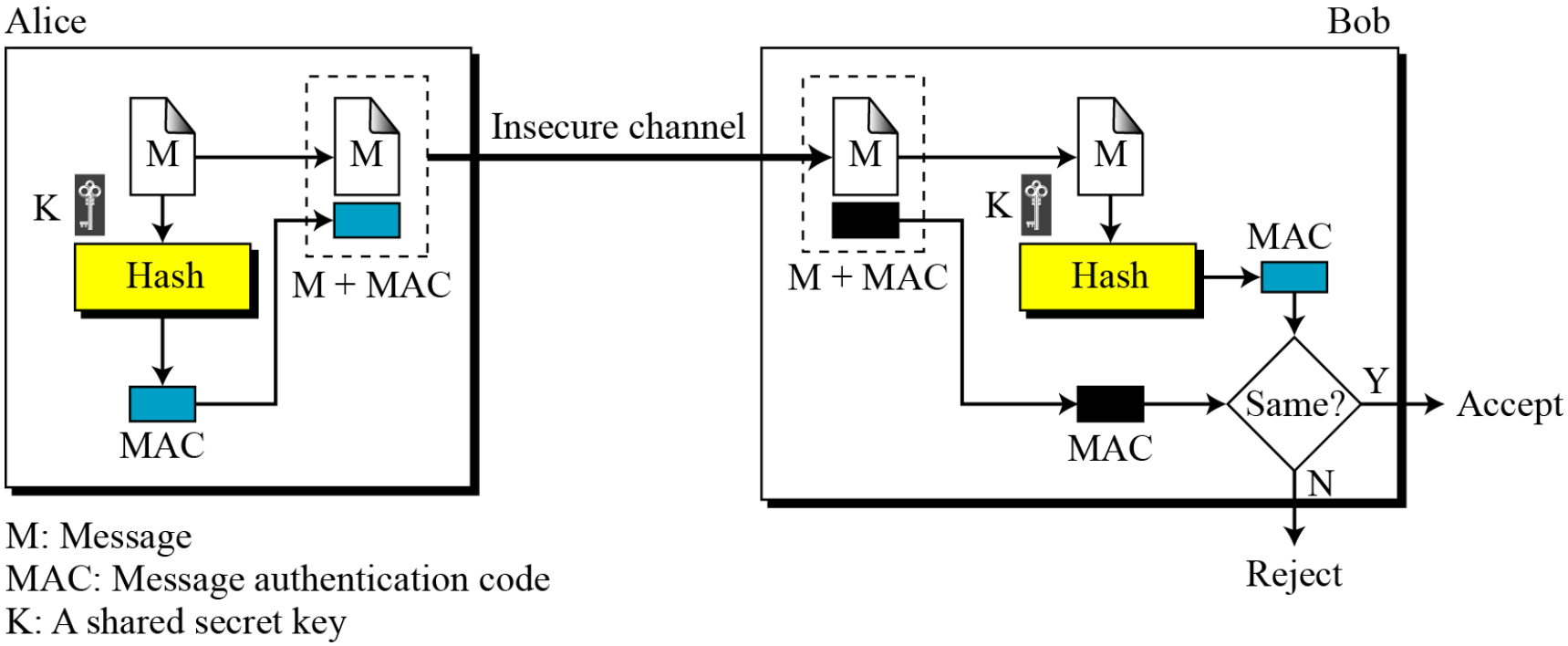
11.3.2 Message Authentication Code (MAC)

Figure 11.9 *Modification detection code (MDC)*



11.3.2 Message Authentication Code (MAC)

Figure 11.10 *Message authentication code*



11.3.2 Continued

Note

The security of a MAC depends on the security of the underlying hash algorithm.

Nested MAC

Figure 11.11 *Nested MAC*