Indeed, some vendors have created special instructions to support AES encryption and decryption [832, 833, 1242].

Like the DES, the AES has spurred studies in cryptanalysis. One effect of these studies is a deeper understanding of how block ciphers work, how to cryptanalyze them, and how to design them to resist attacks. Other effects of these studies remain to be seen.

10.3 Public Key Cryptography

In 1976, Diffie and Hellman [564] proposed a new type of cryptography that distinguished between encipherment and decipherment keys. One of the keys would be publicly known; the other would be kept private by its owner. Symmetric cryptography requires the sender and recipient to share a common key. Public key cryptography does not. If the encipherment key is public, to send a secret message simply encipher the message with the recipient's public key. Then send it. The recipient can decipher it using his private key. Chapter 11, "Key Management," discusses how to make public keys available to others.

Interestingly, James Ellis, a cryptographer working for the British government's Communications-Electronics Security Group, developed the concept of public key cryptography (which he called "non-secret encryption") in a January 1970 report. Two of his colleagues found practical implementations. This work remained classified until 1997 [629].

Because one key is public, and its complementary key must remain secret, a public key cryptosystem must meet the following three conditions:

- It must be computationally easy to encipher or decipher a message given the appropriate key.
- It must be computationally infeasible to derive the private key from the public key.
- It must be computationally infeasible to determine the private key from a chosen plaintext attack.

The first system to meet these requirements generates a shared session key (see Section 11.2.3.1).

Public key systems are based on hard problems. The first type uses NP-complete problems that have special cases that are easy to solve. The system transforms that simpler problem into the more general problem. The information to do this is called "trapdoor information." If an adversary finds that information, the problem can be transformed back into the simpler one, and the adversary can break the system.

EXAMPLE: An early public key cipher was based on the knapsack problem. Given a set of numbers $A = \{a_1, \ldots, a_n\}$ and an integer C, find a subset of

A whose integers add exactly to C. This problem is NP-complete. However, if the a_i are chosen so that each $a_i > a_{i-1} + \cdots + a_1$, then the knapsack is called *superincreasing* and can easily be solved. Merkle and Hellman [1324] developed trapdoor information allowing them to construct a trapdoor knapsack from a superincreasing one.

In 1982, Shamir developed a polynomial-time method for determining trapdoor information [1723], thereby breaking the knapsack cipher. In 1984, Brickell extended this by showing how to break a cipher consisting of iterated knapsacks [293].

A second type is based on hard mathematical problems such as finding the factors of a very large number. The RSA cryptosystem (see Section 10.3.2) provides confidentiality, authentication, and integrity using a problem related to factoring.

An important comment about the examples in this section is necessary.

In the examples that follow, we will use small numbers for pedagogical purposes. In practice, the numbers would be much larger, and often the encipherment schemes will use additional techniques to prevent the success of attacks such as precomputation (see Section 12.1.1) and changing the order of the ciphertext blocks (see Section 12.1.2).

10.3.1 El Gamal

The El Gamal cryptosystem [627] provides message secrecy. It is based on the discrete logarithm problem.

Definition 10–2. Let n, g, and b be integers with $0 \leq a < n$ and $0 \leq b < n$. The *discrete logarithm problem* is to find an integer k such that $0 \leq k < n$ and $a = g^k \bmod n$.

Choose a prime number p with $p-1$ having at least one large factor. Choose some g such that $1 < g < p$; g is called a *generator*, because repeatedly adding g to itself, and reducing mod p, will generate all integers between 0 and $p-1$ inclusive. Next, select an integer k_{priv} such that $1 < k_{priv} < p - 1$, and take $y = g^{k_{priv}} \bmod p$. Then k_{priv} will be the private key and the triplet $K_{pub} = (p, g, y)$ will be the public key.

EXAMPLE: Alice chooses $p = 262643$, a prime number; $p - 1 = 262642 = 2 \times 131321$ has at least one large factor, so her choice is suitable. She chooses $g = 9563$ and the public key $k_{priv} = 3632$. Then

$$y = g^{k_{priv}} \bmod p = 9563^{3632} \bmod 262643 = 27459$$

so the public key is $K_{pub} = (p, g, y) = (262643, 9563, 27459)$.