Public Key Cryptography

Difficulties with Private Key Systems

The main difficulty with the conventional (private key) systems is ***key distribution*** :- *how to get the sender and receiver the same secret key*. In general, doing this is either very expensive or impossible. In a large network using secure messaging, each pair of participants needs to have their own secret key. The distribution and management of this set of keys is a nightmare.

In situations where two parties want to communicate in secret or not (through their computers) and are unknown to each other (say a business and a new customer) there is no easy way to verify the identities of each party to each other. This is known as the **authentication problem**, and what is needed is a means of providing a verifiable "*digital signature*".

Public Key Systems

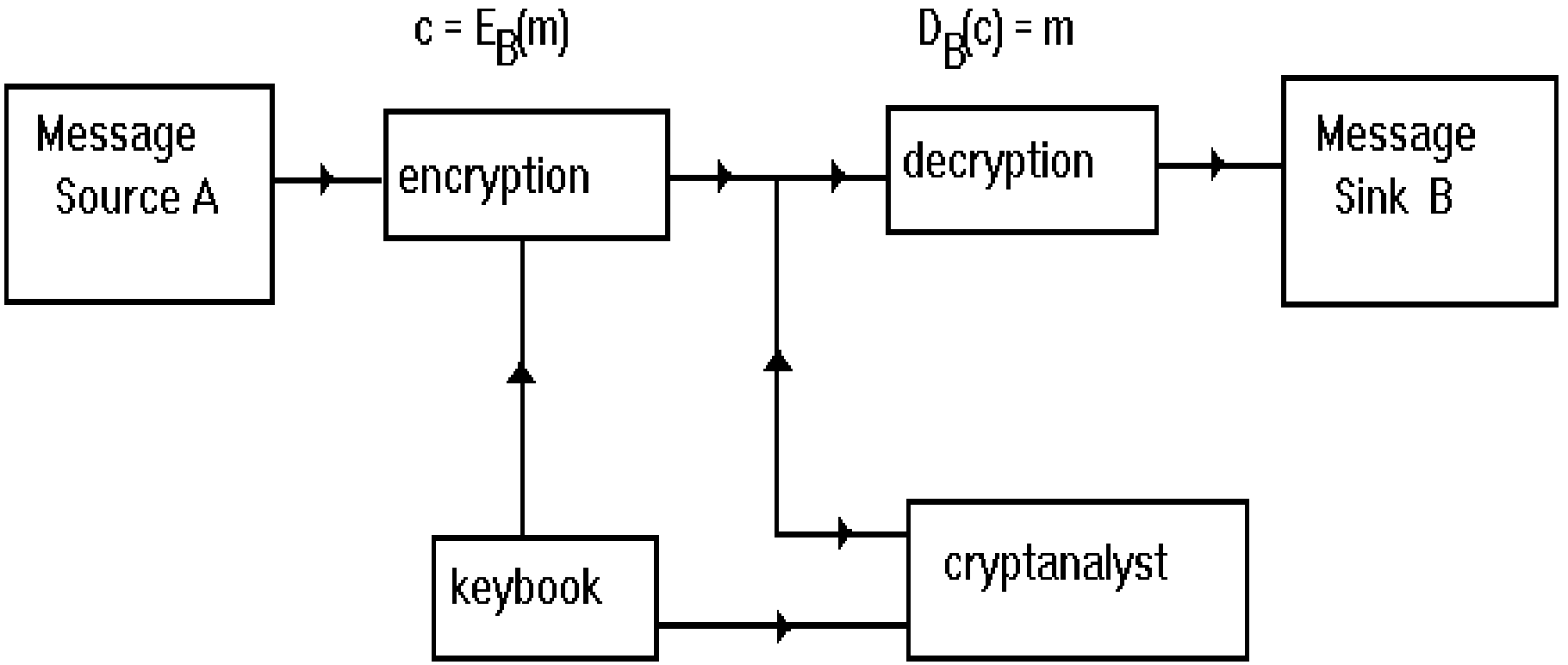
Diffie and Hellman (1976) introduced the idea of a ***public key cryptosystem.***

The system is based on each participant having a *public encryption algorithm* (or a common algorithm and a public key) E_U and a *private decryption algorithm* (or a common algorithm with a private key) D_U. These algorithms are inverses in the sense that:

PK1: $D_U(E_U(m)) = m$ for every message m and user U.

The E_U are made public and are available to everyone.

Public Key Systems



How It Works

A sends a message m to B: A looks up E_B and sends $c = E_B(m)$. Upon receiving the message, B applies the secret $D_B(c) = m$. Since B is the only one who has D_B, B is the only one who can read this message.

PK2: The algorithms do not need much computing time nor memory storage.

PK3: It is practically impossible to find an algorithm D* from knowledge of E_U so that $D^*(E_U(m)) = m$ for all possible m.

Notice that PK3 requires the system to withstand a **chosen text cryptographic attack**.

Trap Door Functions

Diffie &Hellman suggested the use of ***trapdoor one-way functions*** for the encryption algorithm.

A *one-way function* is a function f that is easy to evaluate, but whose inverse f^{-1} is difficult to compute. A ***trapdoor one-way function*** is a one-way function whose inverse is easy to compute given certain additional information.

D&H seemed to have some difficulty coming up with examples, but a few years later a number of systems were suggested. **RSA**, **Knapsack**, etc.

A one-way function can be used for storage of password authorization in a computer.

Signature Schemes

To design a signature protocol one would need:

PK4: $E_U(D_U(m)) = m$ for all messages m and users U.

In order to prevent counterfeiting, we require:

PK5: It is practically impossible to find an algorithm D* from knowledge of E_U so that $E_U(D^*(m)) = m$ for all possible m.

If A wants to sign a message m being sent to B, then A sends $D_A(m) = c$ and B looks up the public encryption key for A and applies it to get $E_A(c) = m$. As D_A is secret, only A could have sent this message ... but anyone can read it.

To send a signed crypted message, A sends $E_B(D_A(m))$. Only B can now read the message.