1 Asymmetric Ciphers

Cryptographic systems rely on keys for encryption and decryption. Traditionally, a single key is required to encrypt and to decrypt. In order for the recipient of the encrypted message to be decrypted by the recipient, the key must also be transmitted. However, sending the key over the channel (normal channel) where the actual message will be sent is insecure. The key must be transmitted on a different and secure channel (key channel)[4]. This secure channel where the key should be transmitted cannot be used for normal transmission because it is costly and sometimes difficult for users to access and use[4]. This begs the question whether it is possible to send encrypted messages in such a way that the key can also be transmitted over the normal (insecure) channel and still achieve secure communication. In this section, we focus on solving this problem by describing the relevant and important work on asymmetric ciphers.

1.1 Merkle(1978)[4]

Secure communication, as described by Merkle[4], allows two parties to communicate in a private manner even though a third party tries its best to learn what is being communicated. We refer to the two parties as Froi and Shiela, and the third party as Jade. Since the key channel is important, the following describes the characteristics of the channel in relation to Jade.

1. All attempts by Jade to change the messages on the key channel are detectable.

2. Jade will not be able to know the actual content of any message passing on the key channel.

The approach by Merkle relaxes the second condition: It is not necessary for Jade not to know what is being sent in the key channel, he can even know everything passing on it. The challenge then is how to securely distribute the key satisfying the conditions above. If Froi and Shiela have agreed upon a key, and the work needed by Jade to find the key is much higher than the effort by Froi and Shiela needed to generate the key , then it is a solution. The effort by Jade should be exponentially higher compared to the effort by Froi or Shiela for a method to be considered a solution.

Merkle's method uses the concept of puzzles[4]. A puzzle is a cryptogram that is meant to be solved. Any encryption function can be used to generate a puzzle. To allow the puzzle to be solved, the key size (N) used in the encryption function is restricted. The difficulty of solving a puzzle can be controlled by adjusting the size of N. A very large size (in bits) of N will make it very difficult to solve the puzzle. In addition, in order to be able to solve the puzzle, some redundancy is needed. Redundancy is introduced by encrypting, along with the original message, some constant known to Froi, Shiela, and Jade. The absence of the constant when a puzzle is decrypted would mean that a wrong key has been used.

Let us consider the scenario when Shiela wishes to send a message to Froi. First, they both agree on the value of N to use. Shiela then generates N puzzles and transmits these N puzzles to Froi using the key channel. Each puzzle generated will have a puzzle ID and puzzle key. The puzzle ID uniquely identifies each puzzle. The puzzle key on the other hand will be used in future communications that will happen once this puzzle has been solved.

When Froi receives the N puzzles, he selects a puzzle at random and attempts to solve the puzzle, with the amount of effort required, as defined by the size of the key space specified by Shiela. After solving a puzzle, Froi sends the puzzle ID back to Shiela using the key channel. The puzzle key, associated with the puzzle ID sent by Froi, is then used for future communications, this time over the normal channel. At this point Jade knows the puzzle ID, since it was sent using the key channel, but not the puzzle key. If Jade wants to know the key, then he must solve puzzles randomly and check the puzzle ID if it matches the one sent by Froi back to Shiela. This will take Jade a long time to solve. To put it formally, Jade will require $O(N^2)$ effort to determine the key whereas Froi will only need, on the average, $O(N)$. The function below generates the puzzles sent by Shiela to Froi. The encryption function is arbitrary.

```
void generate_puzzle()
{
   bit_string id, key, c, random_key, puzzle, k1, k2;
   int i;

   k1 = rand(MAXINT);
   k2 = rand(MAXINT);
   c = rand(MAXINT);
   send(c);
   for (i=0; i<N; i++)
   {
      id = encryption_function(k1,i);
      key = encryption_function(k2, i);
      random_key = rand(c*N);
      puzzle = encryption_function(random_key,id,key,c);
      send(puzzle);
   }
}
```

The code below is executed on Froi's side.

```
void get_id()
{
   bit_string id, key, c, selected_puzzle_id, the_puzzle, current_puzzle,
              temp_constant;
   int i;

   selected_puzzle_id = rand(N);
   receive(c);
   for (i=0; i<N; i++)
   {
      receive(currrent_puzzle);
      if (i == selected_puzzle_id)
         the_puzzle = current_puzzle;
   }
```

```
    for (i=0;i<(c*N);i++)
    {
        id = get_id(finverse(i, the_puzzle));
        key = get_key(finverse(i, the_puzzle));
        temp_constant = get_constant(finverse(i, the_puzzle));
        if (temp_constant == c)
            send(id);
    }
}
```

Once Shiela receives the the puzzle ID from Froi, then the following code will be executed. Key will be used for subsequent communications between the two.

```
void continue_transmission()
{
    receive(ID);
    key = encryption_function(k2, ID);
}
```

The approach by Merkle requires an effort of $O(N^2)$ from Jade to get the key. However, in todays available computing resources, this can be easily broken. The possibility of exponential methods will be more attractive. Also the amount of information sent during the initial setup of the communication is large because N puzzles, consequently N keys, are sent initially.

1.2 Diffie-Helman(1976)[1]

The work by Diffie and Helman[1] proposed a method such that only one “key” needs to be exchanged and in addition the time required from Jade to perform cryptanalysis is exponential. In addition, it allows authentication because its use allows it to be tied to a public file of user information. Shiela can authenticate Froi and vice versa.

Diffie and Helman differentiate public key cryptosystems and public key distribution systems. We let K be the finite key space from which keys K can be obtained and M be the finite message space where messages M are derived. A public key cryptosystem is a pair of families of algorithms E_k andD_k which represent invertible transformations[1].

$$E_k : \{M\} \rightarrow \{M\}$$

$$D_k : \{M\} \rightarrow \{M\}$$

1. for every key K, E_k is the inverse of D_k,
2. for every K and M, the algorithms E_k and D_k are easy to compute,
3. for almost every K, each easily computed algorithm equivalent to D_k is computationally infeasible to derive from E_k,
4. for every K, it is feasible to compute inverse pairs E_k and D_k from K.

Property 3 allows E_k to be made public without compromising D_k.

1.3 Rivest-Shamir-Adleman(1978)[5]

1.4 Elgamal(1985)[2]

1.5 Elliptic Curve Cryptosystems(1987)[3]

References

[1] Whitfield Diffie and Martin Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976.

[2] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *Information Theory, IEEE Transactions on*, 31(4):469–472, July 1985.

[3] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[4] Ralph C. Merkle. Secure communications over insecure channels. *Commun. ACM*, 21(4):294–299, 1978.

[5] Ronald L. Rivest, Adi Shamir, and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.