Applied Cryptography – CSE 539

Karthik Srinivaasan Ayengar Devanathan
School of Computing, Informatics and Decision Systems Engineering
Arizona State University
kayengar@asu.edu

I. Introduction

The course consisted of six assignments which together constitutes this portfolio report. The course focuses on various cryptographic techniques and covers a broad spectrum of topics such as Symmetric key cryptography, public key cryptography, hashing, digital signatures, rainbow tables and certificates.

Cryptography is critical today, especially when communicating on public networks. It makes sure that any message sent, even though visible to others, is completely converted into gibberish using a key, and is decrypted back to the original message at the other end.

Let us see the basic working of a public key cryptographic function and a symmetric cryptographic function:

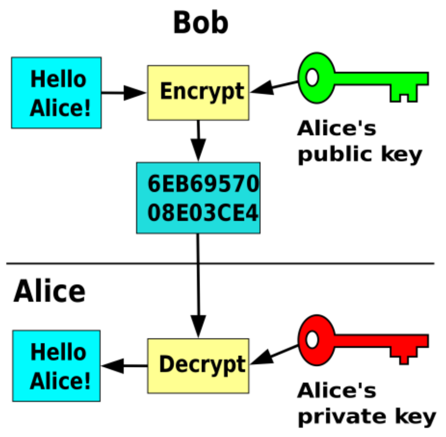

Fig 1. Public key Cryptographic function

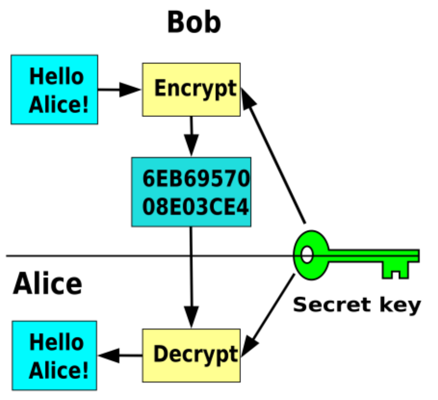

Fig 2. Symmetric Cryptographic Function

As you can see from the above figures, the cryptographic function is public to everyone, including the attackers. The only thing which should be hidden in the function is the key.

In recent times, Cryptography is based on Mathematical Algorithms, especially using large prime numbers so that it becomes nearly impossible for the attacker to factorize and find the key.

Hashing is another part of Cryptography and it is often confused with Encryption itself. Hashing is a one-way function and the plain text cannot be retrieved back the way it is obtained by applying the decryption function. This is especially useful when passwords are to be stored in databases. Any password that the user inputs is hashed and checked against the stored hash value in the database because the Hash function always gives the same output every time. There is a concept called Hash Collision where 2 different inputs give out the same hash value. A good hash function should make sure that there is no or minimum number of collisions.

Digital Signatures are one of the major applications of Cryptography. A digital signature is nothing but a message which is hashed using a known hash function and encrypted using the signer's private key. This allows anybody to use the signer's public key and decrypt it, and hash the message which they receive and see if they match. This is a good authentication scheme since it allows the receiver to verify that it is the right person sending the doc.

There are various other cryptographic applications and algorithms that I studied in the course including secure digital elections, bitcoins etc. As much as it is was fun learning these algorithms, it is also critical for Software Developers to have a knowledge about cryptographic systems.

II. Explanation of the Background and Solution

The first assignment was to design a 32-bit encryption algorithm. The algorithm that we designed uses several already known techniques to induce obfuscation and arbitrariness. Let us look at the following figure to understand the steps that the plain text has to undergo before it becomes cipher text.

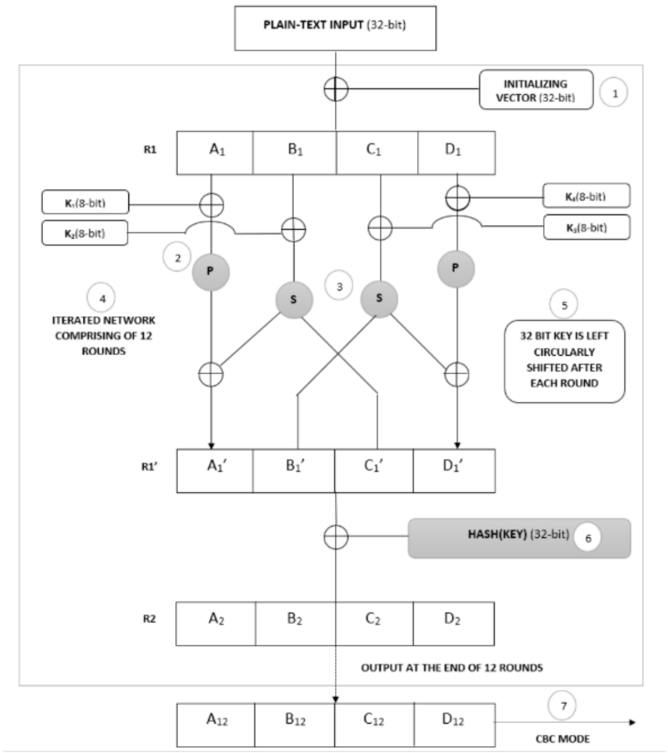
PLAIN-TEXT INPUT (32-bit)
INITIALIZING VECTOR (32-bit)
1
R1
A1
B1
C1
D1
K1(8-bit)
K2(8-bit)
K4(8-bit)
K3(8-bit)
2
P
S
3
S
P
4
ITERATED NETWORK COMPRISING OF 12 ROUNDS
5
32 BIT KEY IS LEFT CIRCULARLY SHIFTED AFTER EACH ROUND
R1'
A1'
B1'
C1'
D1'
HASH(KEY) (32-bit)
6
R2
A2
B2
C2
D2
OUTPUT AT THE END OF 12 ROUNDS
A12
B12
C12
D12
7
CBC MODE

Fig 3. Encryption Algorithm

- The intitializing vector is used to make sure that there is no recurring patterns.
- Once the message is XORed with Initializing vector, it passed through a Permutation-Box for the purpose of diffusion.
- After that, the S-Box operates on it for inducing confusion.
- The key is hashed and is XORed with the resulting cipher from the previous step.
- It constantly undergoes circular shift to make use of all the bits of the key.

The basic algorithm that it follow is CBC chaining. The output of this encryption algorithm was a complex cipher that doesn't make any sense to humans. Attacking this algorithm using brute-force would be very expensive and it is impossible to break it.
There are not many shortcomings in the algorithm, but one noticeable shortcoming lied in the way we split our key into 4 quarters and XORed it. This might make the attacker to get a piece of plaintext with lesser effort.

The second assignment comprised of 3 small tasks. The first task was to find flaws in the given cryptographic algorithm. The algorithm was not explicitly given, but instead a C program was provided. It is a known fact that the headers of a certain file types can be used to get the size, type and other information about a file. The given algorithm did not make any changes to the first 4 bytes of the files, and stored the size of the file as it is. This is enough information to get the type of the file being sent. The subsequent operations are all XOR operations which means that the plain text value is reversible and it can be got by XORing the result with the 2nd operand.
The second task was to find the key. We were given a pdf file and a png file. As mentioned in the previous section, enough information is contained in their headers compute the key. We were successfully able to find the key by XORing the file type with the header cipher. Then we entered the key and decrypted the whole file. This was not the case for text files as the file type is not mentioned in the headers. But, what helped us to find the key for the text file is that they only contain ASCII characters. Therefore, we try all the combinations of key and try till we get a plaintext with all ASCII values.

The third assignment was to crack passwords. We know that authenticating with the help of a text password is the most popular authentication method. But, cracking them is relatively easier compared to other methods because humans are prone to keep combinations of words from dictionary. In this assignment, we were given a set of hashes. We had to find the password from there. There were two ways to achieve this. One was using the traditional brute force method and the other was using rainbow tables. Let us have a look at both the approaches in detail:
In the Brute Force method, we try out every combination of 4 letter words and we hash them to check if any of these values collide with the given hash value. Since we assume that a strong hash function is used, we also assume that there is only one collision for a given password. The below is a snippet of the C program used for this assignment:

```
char i,j,k,l;
 for(i=0;i<=127;i++){
    for(j=0;j<=127;j++){
         for(k=0;k<=127;k++){
             for(l=0;l<=127;l++){
             if(i.strcat(j).strcat(k).strcat(l).strcmp(hash)==0)
             printf(hash);
             break;
             }
         }
      }
}
```

The second method is by constructing a rainbow table. We construct it by using stolen hashes. We alternate between the two steps: reduction and hashing. Each reduction gives a valid password. That password gives some more hashes which in turn gives more reductions. This is repeated until there is a hash which matches a hash in the rainbow table. If there is a collision, then that hash is reduced until it reaches the pre-image of that hash. This allows us to find the actual value much faster than the brute force method.

The fourth assignment dealed with SSL certificates. The intent of the assignment was to help us gain a deep understainding about SSL certificates. This is a way of authenticating the sender without digital signatures. Most of the websites today have a valid certificate, with which we identify its validity.
The first task was decrypt and then encrypt the data using private key and then public key. The second task was to retrieve the public key and private key from the certificate. We used some of Java's basic library to achieve this. The next task was the get the digital signature from the certificate and display it. The last task was to find whether a given certificate was a valid certificate or not. We found that the certificate which was given to us was a self signed certificate.

The fifth assignment hepled us gain understanding about the famous Cryptographic algorithm RSA. We know that RSA works well with the help of Big Prime numbers.
The first task was to try out RSA with small primes, one prime and one composite, two composite numbers etc. We noted how strong the algorithm was with each of these numbers. We noticed that the encryption scheme was prone to atatcks when composite values were used because they can be factorized easily. They proved to be good when big primes were used.
The second task was to verify a challenge-response. We verified that the algorithm works by encrypting and sending it to another user and then decrypting it on the other side to retrieve the original message.
The third task was to implement a blind signature scheme. We implemented it by multiplying a encrypted message with a constant and sending it to a recipient. Without seeing it, the authority signs it blindly. Now we have a valid signed message which we can use. This is just one implementation of a blind signature.

III. Contributions to the Project

All the assignments were done as a team of two. Both of us had brainstorming sessions before beginning the assignments. The results were all documented and submitted in time. Let us look at how the work was split for each assignment.
For the first assignment, we both read on different chaining techniques. We then discussed the limitations and advantages of different techniques and decided on CBC and implemented a cryptographic algorithm.
The second, third and fourth assignment were done in a similar fashion. The fifth assignment, we divided the tasks equally amongst ourselves. I was responsible for verifying the certificate and for extracting the private and public keys while Anand was responsible for retrieving the digital signature and for building the verification functions.
In the last assignment, I was responsible for checking RSA algorithm with different numbers, and my teammate was responsible for implementing blind signatures. We both implemented the challenge-response scheme and verified the communication.
We equally contributed to every assignment by using version control systems

IV. Skills, techniques, and knowledge gained

I gained a lot of knowledge about cryptography and security. I learnt about the various algorithms used for encryption and decryption. As mentioned in the introduction, I learnt about block ciphers, stream cipher, symmetric key cryptography, public key cryptography and a lot of other interesting concepts.

Since the assignments were done in one of the object-oriented programming languages, I became more proficient in C, C++ and Java. In fact, this course has helped me to learn and use different cryptographic libraries in these languages. The course also helped taught me the various applications of cryptography and how to defend against security attacks. In fact, the assignments and exams were focused on finding vulnerabilities and fixing them, which are crucial things to understand in a world of security breaches.

V. References

[1] Settia, Neetu. "Cryptanalysis of modern cryptographic algorithms." IJCST 1, no. 2 (2010).
[2] Schneier, Bruce. Applied cryptography: protocols, algorithms, and source code in C. john wiley & sons, 2007.
[3] Figure 1: https://upload.wikimedia.org/wikipedia/commons/thumb/2/27/Symmetric_key_encryption.svg/2000px-Symmetric_key_encryption.svg.png
[4] Fig 2: https://upload.wikimedia.org/wikipedia/commons/f/f9/Public_key_encryption.svg
[5] Cryptography: https://en.wikipedia.org/wiki/Cryptography
[6] Public Key Cryptography: https://en.wikipedia.org/wiki/Public-key_cryptography
[7] Blind Signatures: https://en.wikipedia.org/wiki/Blind_signature