CALIFORNIA STATE POLYTECHNIC UNIVERSITY, POMONA

CRYPTOGRAPHIC ALGORITHMS (AES, RSA)

A PAPER

SUBMITTED TO

PROFESSOR GILBERT S. YOUNG

IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE COURSE

CS530 : ADVANCED ALGORITHM DESIGN AND ANALYSIS

BY

HESHAM DARWISH

010939825

POMONA, CALIFORNIA

DECEMBER 3, 2015

ABSTRACT

Cryptography is the use of codes and ciphers to protect secrets, and it began thousands of years ago within and between governments and military forces. But after the internet was introduced in 1990s, cryptographic algorithms and protocols became necessary and crucial to keep systems secure and keep the private information of every person on the internet secure. This paper will discuss the purpose and types of Cryptographic algorithms. And then will go in details of two of the widely used cryptographic and encryption algorithms that are mainly used to ensure data communication security; RSA Algorithm which was published in 1977 by MIT students and AES Algorithm which was introduced in 2001 after winning the NIST competition and has been adopted since then by the US government as the standard encryption technique.

I. INTRODUCTION

Cryptography is the use of codes and ciphers to protect any information, it’s first known use of cryptography was 1900 BC, when one of the egyptian pharaohs used a type of a non standard hieroglyphs in an inscription. In the early 20th century, it became much more complicated and sophisticated with the use of mechanical and electromechanical machines. In the 1970s, cryptography took another major advance to be linked with encryption which is the conversion of information from a readable state (plaintext) to nonsense (ciphertext) in a way that only authorized parties can read it. This done by applying algorithms (Mathematical formulas and functions) on the data. Encryption basically denies the message content to any interceptor who wants to access the data. There are two main types of encryption techniques, symmetric key encryption and asymmetric key encryption. In symmetric key encryption, the sender uses a key to encrypt the data and send it to the receiver who will use the same key to decrypt the data, while the challenge is to securely transfer the key that will be used in the process. In asymmetric key encryption, each of the sender and the receiver have two set of keys; a public key which is shared publically with everyone, and a private key that is kept private and it’s used to decrypt any message encrypted by the public key. This paper will go through two different algorithms that is widely used nowadays for encryption. Advanced Encryption Standard (AES) based on Rijndael Algorithm, and RSA Algorithm . AES is the US encryption standard and it’s based on the symmetric key encryption technique, and its the most important cryptographic algorithm in the world of cybersecurity. While RSA is based on asymmetric key encryption and its heavily used for ensuring data integrity and authentication.

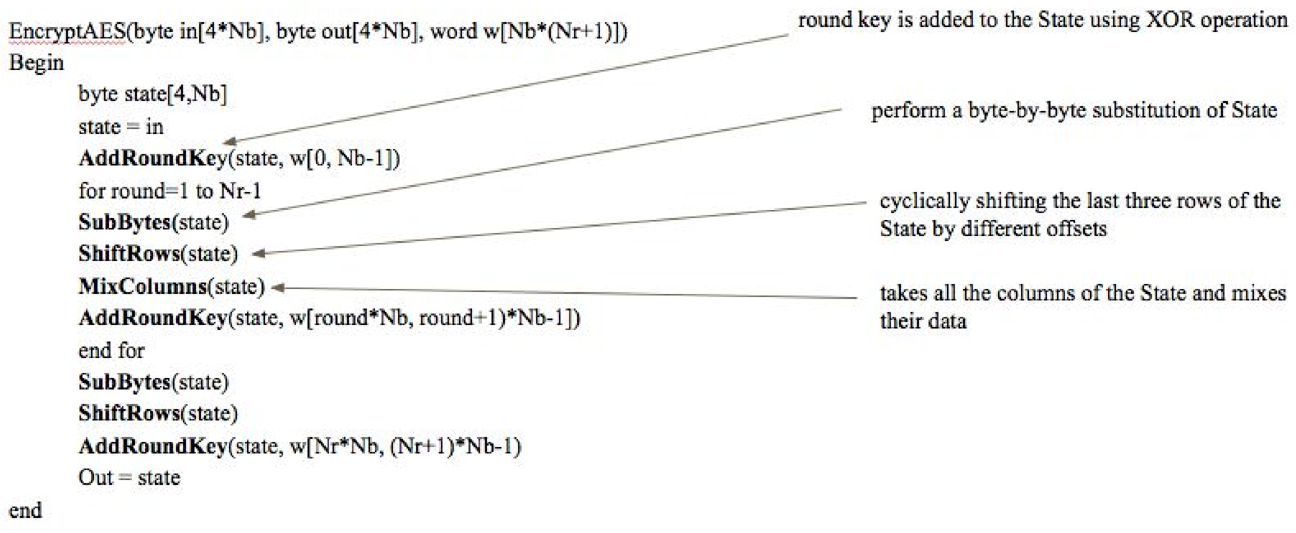
II. AES (Advanced Encryption Standard)

A. Introduction

The Advanced Encryption Standard (AES) original name was Rijndael, as it was named after the developers of the algorithm; Vincent Rijmen and Joan Daemen. The Algorithm was submitted as a proposal to the NIST (National Institute of Standards and Technology) in 1996 during a competition done to select an encryption standard to replace the existing one at that time DES (Data Encryption Standard) which was published in 1977. The NIST committee announced the AES as the US standard in November 26, 2001. AES uses symmetric key encryption where the same key is used for encrypting and decrypting the data, and the main challenge is to exchange that key with complete privacy as if this key is found then all the encryption process is compromised and useless.

B. Algorithm Explanation

Algorithm Pseudo Code



```
EncryptAES(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
Begin
        byte state[4,Nb]
        state = in
        AddRoundKey(state, w[0, Nb-1])
        for round=1 to Nr-1
        SubBytes(state)
        ShiftRows(state)
        MixColumns(state)
        AddRoundKey(state, w[round*Nb, round+1)*Nb-1])
        end for
        SubBytes(state)
        ShiftRows(state)
        AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1)
        Out = state
end
```

round key is added to the State using XOR operation

perform a byte-by-byte substitution of State

cyclically shifting the last three rows of the State by different offsets

takes all the columns of the State and mixes their data

The AES Algorithm is based on a design principle known as a substitution-permutation network, which is basically a combination between substitution and permutation. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. The key size used for an AES cipher specifies the number of transformation rounds used in the encryption process. The number of rounds are as following:

- 10 rounds for 128-bit keys.
- 12 rounds for 192-bit keys.
- 14 rounds for 256-bit keys.

Each round consists of several processing steps, each containing four similar but different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to decrypt the data (transform ciphertext back into the original plaintext) using the same encryption key. The encryption key used in this process is originally 16 bytes, so it is used in key expansion algorithm to generate a number of subkeys where each subkey is used in each round in the encryption process. The original cipher key is expanded from 16 bytes to 16 (number of rounds +1) bytes and then broken into round keys where a round key is needed after each round and before the first round. The details of the key expansion algorithm are complex and out of the scope of this paper.

There are four main functions in this algorithm:

1. **Add Round Key**

In this function, the subkey is combined with the state, and this happens by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.

2. **SubBytes**

In this function, each byte in the state block is substituted with a sub-byte from Rijndael S-box which is a block that is constructed by combining the inverse function with an invertible affine transformation. This step provides the non-linearity in the cipher.

3. **ShiftRows**

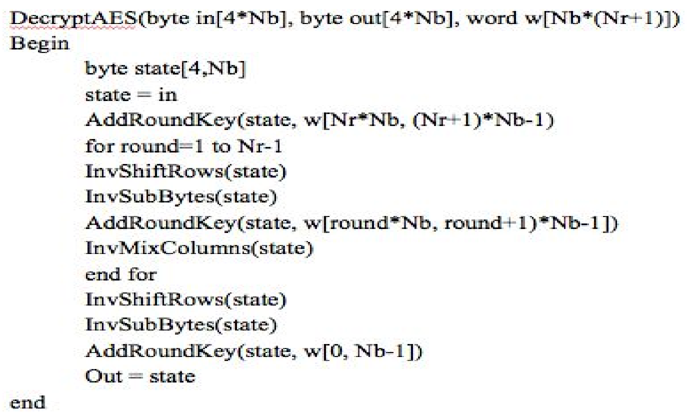
In this function, an operation is done on the rows of the state; it cyclically shifts the bytes in each row by an offset. The first row is left unchanged, the second row is shifted one to the left, the third is shifted by two to the left, and the fourth is shifted by three to the left. The rule here is basically the row n is shifted by n-1. This step provides assurance that the columns will not be linearly independent.

4. **MixColumns**

In this function, the four bytes of each column of the state are combined using a linear transformation, where the function takes four bytes as input and outputs four bytes where each input byte affects all four output bytes. This is step is considered the most important step for the diffusion in the cipher.

C. Decryption Algorithm

Decryption Pseudo Code



```
DecryptAES(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
Begin
        byte state[4,Nb]
        state = in
        AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1)
        for round=1 to Nr-1
        InvShiftRows(state)
        InvSubBytes(state)
        AddRoundKey(state, w[round*Nb, round+1)*Nb-1])
        InvMixColumns(state)
        end for
        InvShiftRows(state)
        InvSubBytes(state)
        AddRoundKey(state, w[0, Nb-1])
        Out = state
end
```

Decryption Explained

Decryption is considered really simple after understanding the encryption process, it is basically the inverse. The algorithm was designed in a way that makes all the steps to be invertible. The important aspect here is that the same key has to be used because the AES is considered a symmetric key algorithm, and this original key will also be used in the key expansion algorithm to generate all the round keys that will be used in the decryption process. The decryption starts at the last round and the last round key, so the round key is added first to the last round, then the MixColumn step is applied, after that the shifts are done backwards as well. In ShiftRow step instead of shifting left, we will shift right. Lastly the subBytes is applied using the inverse S-box, which can be done easily by shifting the rows and columns in the original S-box. After all the rounds have been completed in the opposite order, the final state will contain the original plaintext.

III. RSA

A. Introduction

RSA was introduced in the fall of 1976 in MIT, when the computer scientists Ronald Rivest and Adi Shamir, along with the mathematician Leonard Adleman were able to come up with the RSA Algorithm where R is for Rivest, S for Shamir, and A for Adleman. Since 1976 and till this day, the RSA is widely used for encryption, key exchange and digital signature. The RSA algorithm was a product that came out after many months of work, where they were trying first to do an encryption technique based on the knapsack problem but everytime Rivest and Shamir come up with an algorithm, Adleman would poke holes in it in few minutes, until one day Rivest came up with an idea and Adleman wasn't able to poke a hole in it, so that was the birth of the RSA algorithm.

B. Algorithm Explanation

The RSA algorithm involves three main steps; key generation, encryption and decryption. RSA is an asymmetric key algorithm where it involves two keys - a private key and a public key. As the name suggest, anyone can be given information about the public key, but on the other hand the private key must be kept secret. The idea of it is that anyone can use the public key to encrypt a message but only the person who own the corresponding private key can decrypt it. The key power and the security of the RSA algorithm is based on the fact that the factoring problem is "hard", where the only way to break the RSA is to find an efficient algorithm for factoring large numbers, and this doesn't exist. On the other hand, it is proven now that it will not be the case when the quantum computers are introduced. So the introduction of quantum computers will probably kill the RSA.

The public and private keys are generated in RSA by following these steps

1. Choose two distinct prime numbers p and q. In order for the system to be secure, the integers p and q should be chosen at random and should be of similar bit-length. To find large primes, the numbers can be chosen at random and, using one of several fast probabilistic methods, we can test their primality.

2. Compute $n = pq$. The product n will be used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the length of the key.

3. Compute $\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1)$, where φ is Euler's totient function.

4. Choose an integer e such that $1 < e < \varphi(n)$ and gcd$(e, \varphi(n)) = 1$ (that is, e and $\varphi(n)$ are coprime). The number e is released as the public key exponent.

5. Determine d as $d \equiv e^{-1}$ mod $\varphi(n)$. That is, d is the multiplicative inverse of e (modulo $\varphi(n)$). This is often computed using the extended Euclidean algorithm. The number d is kept as the private key exponent.

So after doing all the previous steps, the public key is formed by the pair (n, e), where n is called the modulus and e is called the public (or encryption) exponent. The private key is formed by the pair (n, d), where d is called the private (or decryption) exponent. It is imperative that the decryption exponent d is kept secret. In addition, the numbers p, q, and $\varphi(n)$ must also be kept private because they can be used to calculate d.

Once the keys are determined, secure messages can now be sent. So for example if Bob would like to send Alice a message, Alice will transmit her public key (n, e) to Bob, who will convert the message into an integer m, such that $0 \leq m < n$, then bob will be able to compute the ciphertext using the formula $c \equiv m^e$ **(mod n).** After that, Bob sends c (ciphertext) to Alice who will decrypt the ciphertext by using her private key and computing the formula $m \equiv c^d$ **(mod n),** so she will be able to get m which can be transferred again easily to the original message.

C. RSA Example

For p=3, q=11. Generate private and public key for an RSA system and demonstrate encryption and decryption for m=2.

n=pxq=3x11=33, □(n)=(p-1)(q-1)=2*10=20.   we choose e=7   Pu=(7,33), Pr=(3,33)
de mod □(n) =1
d(7) mod 20=1 ---> d=3
Encryption of m=2
C=M^e mod n
C= 2^7 mod 33 = 128 mod 33 = 29, ciphertext=29

So send the encrypted message - ciphertext, 29

Decryption of C=29
M=C^d mod n = 29^3 mod 33 = 2
M=2 -- Original plaintext

D. RSA Logistics

The RSA Algorithm has remained a secure scheme for sending encrypted messages for almost 40 years. And it is still used now for digital signature and Key exchange, it’s mainly used on top of AES or any other mean of encryption to add extra layer of protection or even to be able to exchange the encryption keys. The only way to break the RSA in use today is to factor the modulus n, but there is no polynomial time algorithm for factoring large numbers. Although it’s proven that a quantum computer could be used a factor of n in polynomial time, thus breaking RSA. For now, RSA should remain secure as long as n is sufficiently large, it’s recommended to be at least 2048 bits or 4096.

IV. CONCLUSION

This paper focused on two of the most used cryptographic algorithms in our world today, AES and RSA. This paper discussed the history as well as the algorithm structure and components of AES and RSA and how they exactly work. The AES is a symmetric key cryptographic algorithm that uses an cipher key and the message as an input and outputs a ciphertext which no one can understand or interpret unless have in hand the symmetric cipher key. The AES is a long process which can be 10, 12, or 14 rounds of transformation depending on the size of the key, and it consists of four main functions; AddRoundKey, SubBytes, ShiftRows, and MixColumns. The AES-256 bit is the standard encryption technique worldwide nowadays. On the other hand, RSA is an asymmetric key cryptographic algorithm where each participant have a public key and a private key, and all the public keys of all participants can be shared, so that the sender can encrypt the message with the public key of the receiver so the receiver will be the only one who can decrypt the message. RSA is mainly used nowadays for exchange keys in top of another encryption technique or in digital signatures.

V. REFERENCES

1. Kaufman, C., Perlman, R., and Speciner, M. *Network Security: Private Communication in a Public World*. 2nd ed. Upper Saddle River, N.J.: Prentice Hall PTR, 2002.

2. *Rijndael.Advanced Encryption Standard (AES)*. FIPS. November 23, 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf (accessed December, 5, 2015).

3. NIST (2001). Announcing the Advanced Encryption Standard (AES). http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. (accessed December, 5, 2015)

4. Daemen, J., and Rijmen, V. *AES Proposal: Rijndael*. September 3, 1999. http://www.comms.scitech.sussex.ac.uk/fft/crypto/rijndael.pdf (accessed December, 5, 2015).

5. Rivest, R.; Shamir, A.; Adleman, L. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". MIT. 1977. http://people.csail.mit.edu/rivest/Rsapaper.pdf. (accessed December, 5, 2015).

6. Robinson, Sara. “Still Guarding Secrets after Years of Attacks, RSA Earns Accolades for its Founders.” SIAM News, Volume 36. June 2003.

7. M. J. Wiener, “Cryptanalysis of short RSA secret exponents,” IEEE Transaction on Information Theory, Vol.36, No.3, pp. 553-558, May 1990.

8. Susan Landau, Sun Microsystems. “ A Brief Summary of Attacks on RSA”, http://www.math.duke.edu/~holden/Math65S/attacks-rsa/. (accessed December, 5, 2015).