Authenticated but Repudiable Cryptography

Richard Matthew
School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
richard.matthew20@gmail.com

***Abstract*—Sending secret messages for spies are not easy, there are few problems regarding this message sending. If we want to protect the confidentiality of the messages, sometimes the messages can be traced back to the sender. If we want to protect the anonymity, we can't authenticate the sender. This paper propose a method to send authenticated but repudiable message by modified RSA algorithm.**

***Keywords*—authentication, repudiation, cryptography, RSA**

I. INTRODUCTION

To send confidential messages that are both authenticated but repudiable is not easy. If we use public key cryptography, it's confidential, authenticated but non-repudiable.

In this paper, I propose an algorithm to provide authentication, confidentiality, integrity but repudiable. This algorithm is a modification from RSA, which rely on big prime numbers. The keys will have a one to many relationship.

II. SECURITY ASPECT

Confidentiality is the ability to protect data from those who are not authorized to view it [1]. Confidentiality in this topic means that only authorized people can send and/or received the messages. Integrity is the ability to prevent data from being changed in an unauthorized or undesirable manner [1]. Integrity in this topic means that only authorized people can change or alter the messages. Authentication is method we use to establish a claim of identity as being true [1]. In this paper, someone is authenticated when he/she has the key. Non-repudiation is the ability to prevents someone to deny their identity. In this paper, we want the opposite way, we want the user to be able to deny their identity.

III. RSA

RSA (Rivest, Shamir and Adleman) is one of the first public-key cryptosystems and is widely used for secure data transmission [2]. RSA is based on two large prime numbers.

1. p and q are prime number (private property)
2. n = p . q (public property)
3. $\Phi(n) = (p - 1)(q - 1)$ (private property)
4. e (encryption key) (public property) (with condition greatest common divisor $(e, \Phi(n) = 1)$)
5. d (decryption key) (private property) $(d = e^{-1} \bmod (\Phi(n)))$
6. m (plaintext) (private property)
7. c (ciphertext) (public property)

So public key is are pair of (e,n) and private key is pair of (d,n)

IV. AUTHENTICATED BUT REPUDIABLE CRYPTOGRAPHY

In this algorithm, I propose that we modified RSA algorithm so it's become repudiable. property will be :

1. p and q are prime number (private property)
2. n = p . q (public property)
3. $\Phi(n) = (p - 1)(q - 1)$ (private property)
4. some B (the many key) (private property) (with condition greatest common divisor $(B, \Phi(n) = 1)$)
5. A (the one key) (private property) $(A = B^{-1} \bmod (\Phi(n)))$
6. m (plaintext) (private property)
7. c (ciphertext) (public property)

Because of the property B is relatively prime to $\Phi(n)$, so we could generate many B that have relation to A. So KeyB are pair of (B,n) and keyA is pair of (A,n)

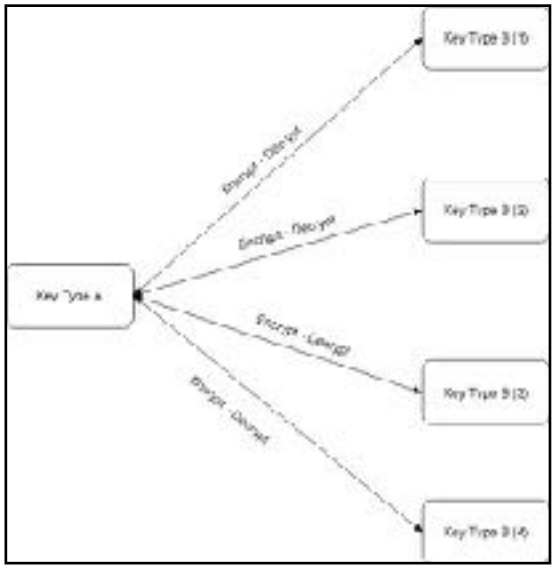

Figure 1. One to Many key relationship

V. USAGE

There's two usages that I can think of:

1. Send confidential but repudiable message
   First sender encrypt plaintext using one of keyB that he/she own. Second we send the ciphertext to receiver. Third the receiver decrypt the ciphertext using his/her keyA. With this kind of scheme, the man in the middle can't figure out the identity of the sender.
2. Send broadcast message
   First sender encrypt plaintext using keyA. Second we

send the ciphertext to receiver. Third the receiver decrypt the ciphertext using his/her keyB.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees and the editor for his valuable suggestions that have resulted in the improvement of the paper.

REFERENCES

1. Andress, Jason. The basics of information security: understanding the fundamentals of InfoSec in theory and practice. Syngress, 2014.
2. https://en.wikipedia.org/wiki/RSA_(cryptosystem) accessed on 10th May 2019.
3. Algoritma-RSA-(2018). 2018. Munir, Rinaldi. ITB