A Study of Identity Based Encryption Systems

Samuel Petit
3rd Year Integrated Computer Science student at Trinity College Dublin, the University of Dublin
Email: petits@tcd.ie

***Abstract*—The abstract goes here.**

I. Introduction

Will be studying IBE. A type of Asymmetric paired keys encryption system. Charcterised with the fact that one of its keys identifies the recipient (email...phone no).

Makes a lot of sense to compare with classical systems currently in use such as RSA.

So let's compare both systems and try to explain why IBE is not as popular

A. Introduction to Public Key Cryptography

Public Key Cryptography, also called Asymmetric Cryptography is an encryption scheme. In other words, it is a method for encrypting messages. Unlike symmetric cryptography which uses the same key for both encrypting and decrypting messages, asymmetric cryptography uses two different keys such that one is used for encrypting and the other for decrypting information. In this context, a key is a string of characters that is used in some mathematical formula to turn a information (a message for example) such that it is impossible to understand in its encrypted form. Similarly, we would then also use a key to map the encrypted information back to its original, usable form. In asymmetric encryption, we typically call the encryption key the Public Key, and the decryption key the Private Key. The first asymmetric key cryptosystem was published in 1976 by Whitfield Diffie and Martin Hellman, previously, all useful modern encryption system used symmetric key encryption systems. While both systems still have their usages in todays world, asymmetric key encryption systems are now used on a daily basis throughout the world. With systems such as both HTTP over TLS and HTTP over SSL protocols, digitally signed files, bitcoin, encrypted messaging services and many other all using some for of asymmetric encryption.

1) How it works: Put simply, let's say we had a public key K_{public} and $K_{private}$. We would then obtain a ciphertext $cipher$ with the following formula:

$$cipher = K_{public}(message)$$

Similarly, we obtain the original message from the ciphertext with the following formula:

$$message = K_{private}(cipher)$$

2) Different implementations of Asymmetric Encryption: Many popular protocols and systems can be used as examples of working asymmetric encryption. To start with, Identity Based Encryption (IBE) uses a set of asymmetric keys. Other popular protocols or systems include the Diffie-Hellman key exchange protocol which is used to exchange cryptographic keys over a non secure channel. RSA is a very popular cryptosystem which includes algorithms and functions to compute a set of keys as well as handling encryption and decryption.

B. Different implementations

There are many different implementations of asymmetric

1) Requirements for public key algorithms (key gen, encrypt and decrypt): explain Requirements here

2) How RSA is Implemented:

$p \leftarrow prime()$ {prime() returns a prime number}
$q \leftarrow prime(N)$
$n \leftarrow pq$
$\phi(n) \leftarrow (p-1)(q-1)$
$e \leftarrow coprime(\phi(n))$ {coprime returns a value coprime to ϕ where $1 < e < \phi$}
$d \equiv e^{-1} mod \phi(n)$

To explain a bit more about this algorithm, we start by generating randomly two prime numbers p and q. We obtain n from their product, n is used as part of the encryption and decryption function as we will see soon. Then we use Euler's totient function such as to count the positive integers up to $(p-1)(q-1)$, we could alternatively use Carmichael's totient function here with a slight change in algorithm, the same keys would be generated regardless. We then pick a value e such that it is positive, smaller than our value obtained from Euler's totient $\phi(n)$ function and coprime to it. We have then obtained the public key $\{e, n\}$. To compute our private key with simply compute the modular multiplicative inverse of e modulo $\phi(n)$ ($d \equiv e^{-1} mod(\phi(n))$) to obtain our private key: $\{d, n\}$.

Given a public key $\{e, n\}$, we can then very simply encrypt a message M. This is done simply by computing:

$$C = M^e \mod n$$

This part is fairly simple as it only contains a single operation, though keep in mind that in practice e could be a very large number so M^e could be demanding on the device.

Finally, for decryption, given that we have a private key $\{d, n\}$, and a ciphertext which was encrypted using the private key's corresponding public key, we can obtain the original

message M using a formula very similar to that of the encryption:

$$M = C^d \mod n$$

RSA is secure given the fact that factoring numbers is complicated and expensive. For instance, let's assume that we have a public key $\{e, n\}$, we could in theory find the values for p and q by factoring n. Recall that $n = pq$. Then all we would have to do is use Euler's theorem to obtain $\phi(n)$ and from there we could find e and finally obtain the private key d from the equation $ed = 1 \mod \phi(n)$.

In practise though, we pick very large values for p and n, factoring very large prime numbers is very hard and trying to obtain p and q using a brute force method would take a very long amount of time. (HOW MUCH ???)

3) Explain how IBE works: explain IBE systems

C. Math Systems at the core of Public Key Encryption

Explain that at the core of public key encryption, underlies many important mathematical concepts.

1) Prime numbers: If there is something you may have noticed, it may be that prime numbers are at the core of many of the concepts being used in the types of asymmetric encryptions we are covering here. There is a reason for that, it is the fact that primes are very easy to multiply together, however factorising a number into two prime numbers is extremely computer intensive. Much more so than it would be if the numbers weren't primes.

So then, how do we know a number is prime ? Do we try all possible factors until we know we have a prime number? This approach would work, however, it would be terribly expensive when we are generating very large primes. Thankfully, there are mathematical theorems we can use to make this check easier.

2) Extended euclidean algo to find gcd, coefficients: euclidean algo

3) Fermat's little theorem: fermats little thm, its place in public crypto

4) Discrete logarithms: discrete log problem

5) Chinese remainder theorem: chinese remainded thm at the core of decryption (or encryption)

6) Square and multiply: square and multiply to compute large exponents

D. What is identity based encryption ?

Explain what is exactly IBE now that we know more about public key encryption

1) IBE typical implementation: how is an IBE system implemented

2) What are its advantages and flaw?: pros and cons of IBE

E. IBE vs RSA

Compare the two

1) Attacks on IBE Public key systems: Go through possible attacks on Public key systems

2) Are these attacks all possible on IBE? is it better or worse with IBE?: go through how IBE protects from attacks differently than rsa systems

3) which is most safe?: expalin which system is safest

II. Conclusion

The conclusion goes here.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.