Abstract

Encryption is a complex and bewildering process, yet it is absolutely foundational for secure and safe activities on the internet. Encryption, in its many forms, ultimately enables identity verification, password protection, secure conversation, cryptocurrency trade, and other online activities. Despite this widespread use, encryption is not a process easily explained to the layperson due to its complexity. Thus, the object of this research is to demystify the process of encryption and provide an understanding of one of the most common forms of modern encryption - RSA Encryption. This will be accomplished through the information provided on my poster as well as an interactive demonstration of my implementation of RSA Encryption. Additional information is included on the topic of Elliptic Curve Cryptography, the other common modern encryption method. With a strengthened understanding of modern encryption, my audience will be empowered to live more security conscious lives.

Encryption Decrypted

What is Encryption?

Encryption is a process by which information is kept secret from those who are not meant to see or modify it. Long ago, the famous "Caesar Cipher" obfuscated top secret military messages by simply replacing every letter with the letter from three positions before it in the alphabet (Cruise, 2014). In the modern day, simple methods like this are well known and offer no protection for sensitive information. Encryption has since evolved into a complex and varied process that relies on the use of a "Trapdoor Function" to quickly generate keys for encrypting and decrypting information in such a way that it would take far too long for another computer to compute the key (Cruise, 2014). Though many methods of encryption exist, two are discussed within the scope of this project - RSA Encryption and Elliptic Curve Cryptography.

Symmetric vs Asymmetric Encryption

Modern encryption methods can be broken down into two main types - symmetric and asymmetric encryption. Imagine for a moment that you and a friend wish to share some secret information. The two of you decide on a secret code and exchange it in a private location or channel. Secrets exchanged through methods similar to this are called symmetric, because both parties share a key, in this case the secret code (Lake, 2018). This method runs into issues if both parties cannot find a private and secure way to exchange the key.

Now, imagine a post office mailbox. Anyone can put mail into the mailbox, but the mail can only be retrieved by someone with a special key. This example demonstrates the basics of asymmetric encryption, which requires two keys. The public key - being able to put mail in the mailbox - is something that anyone can do. However, it can only be undone by someone with a private key, which is not widely distributed. This method of encryption is commonly used across the internet, where two parties may be unable to safely exchange a key for symmetric encryption (Ricks, 2019).

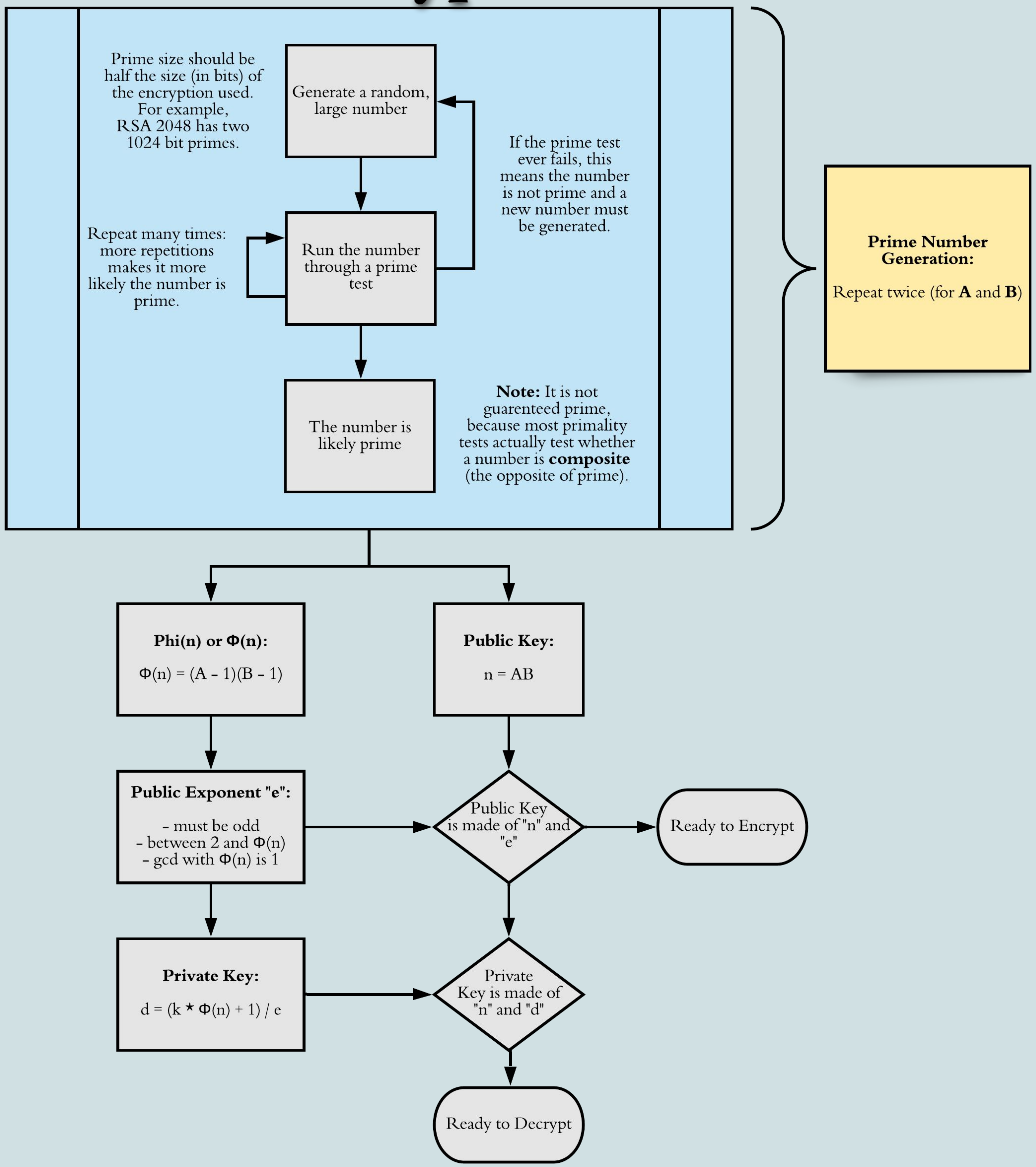
Trapdoor Functions

A Trapdoor Function is a mathematical function which is easy to compute when all necessary information is available, but incredibly difficult otherwise. As an example, RSA Encryption (see graphic) uses large prime numbers to generate a composite number (the public key) and the "phi" (number of integers less than or equal that don't share a common factor) of the composite number. This information is critical to the calculation of the private key. The trick here is that these calculations are trivially easy if the prime numbers are known. However, without this knowledge, brute force methods of testing every possible prime would be necessary. This would take an arbitrarily long time, thus providing a secure private key. That being said, with quantum computing and constantly improving computational power, encryption methods like RSA are expected to be useless before 2030 (Clancy, McGwier, Chen, 2019).

Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) is another commonly used method of encryption. ECC is similar to RSA Encryption in many ways, but it has one major difference. The trapdoor function used in ECC does not rely on prime numbers; ECC repeats the elliptic curve discrete logarithm function some random number of times to generate a key. This turns out to be a highly effective trapdoor function, because even when the start and end state are known, it is very difficult to determine the number of times that the function was repeated. ECC has some major benefits over RSA, namely the much smaller key. As a result of this smaller key, "using elliptic curve cryptography saves time, power and computational resources for both the server and the browser helping [Cloudflare] make the web both faster and more secure" (Sullivan, 2013). Unfortunately, even ECC will be made obsolete by quantum computing.

RSA Encryption



Prime size should be half the size (in bits) of the encryption used. For example, RSA 2048 has two 1024 bit primes.

Generate a random, large number

If the prime test ever fails, this means the number is not prime and a new number must be generated.

Repeat many times: more repetitions makes it more likely the number is prime.

Run the number through a prime test

The number is likely prime

Note: It is not guarenteed prime, because most primality tests actually test whether a number is **composite** (the opposite of prime).

Prime Number Generation:

Repeat twice (for **A** and **B**)

Phi(n) or Φ(n):

Φ(n) = (A – 1)(B – 1)

Public Key:

n = AB

Public Exponent "e":

- must be odd
- between 2 and Φ(n)
- gcd with Φ(n) is 1

Public Key is made of "n" and "e"

Ready to Encrypt

Private Key:

d = (k * Φ(n) + 1) / e

Private Key is made of "n" and "d"

Ready to Decrypt

References

Clancy, T., McGwier, R. W., & Chen, L. (2019, May 15). TUTORIAL: Post-Quantum Cryptography and 5G Security. *WiSec '19: ACM Conference on Security and Privacy in Wireless and Mobile Networks*.

Cruise, B. (Narrator). (2014). *RSA Encryption* [Online video].

Lake, J. (2018, December 10). What is RSA encryption and how does it work?. In *comparitech*.

Ricks, B. (2019). *CSCI3550: Encryption*, week 9, notes [Lecture]

Sullivan, N. (2013, October 23). A (Relatively Easy To Understand) Primer on Elliptic Curve Cryptography. In *Cloudflare*.