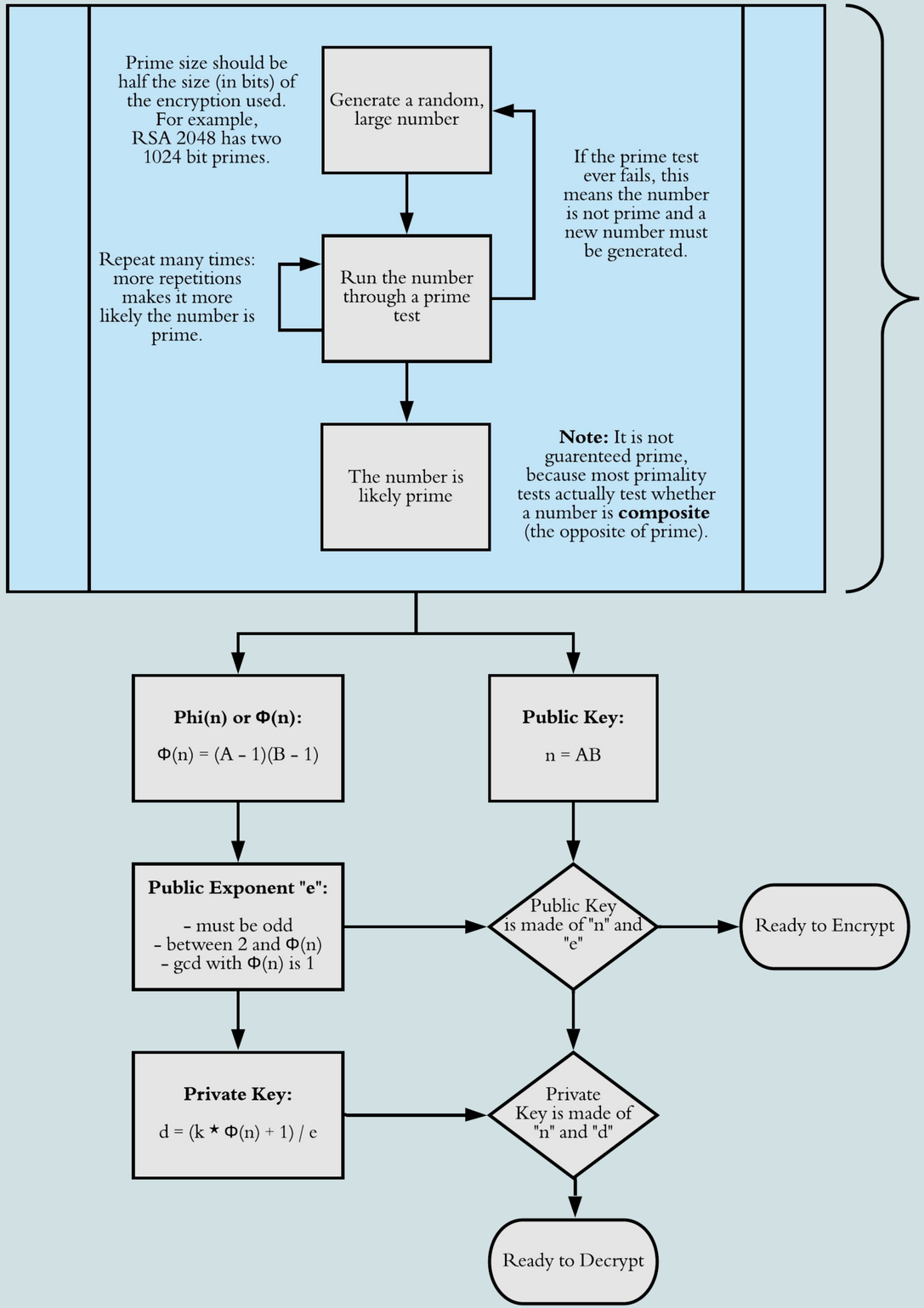
Encryption Decrypted



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. 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 “crack the code.” 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” (the secret code). 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.

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.

Note to the reader: In addition to the information provided here, I wish to write a section about The Crypto Wars, in which the US government has tried to get a “backdoor” into encrypted systems as well as a section on quantum computing and elliptic curve cryptography. I have not yet completed my research on these topics, and they will be included in my final draft of this poster. I have also programmed an implementation of RSA Encryption as a part of my research. This is nearly complete, but I need to put it online so that I can present it alongside this poster.

References

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*.