VAStar Cyber Security

Modern Cryptography

Goals

- Learn about symmetric encryption
- Learn about key sizes
- Learn about asymmetric encryption
- Learn about modular arithmetic
- Learn how modular arithmetic makes public key cryptography possible
- Learn about TLS and its use in internet transactions

Symmetric Encryption

Symmetric encryption is the term for encryption where the key used to encrypt and decrypt the message is the same. Similar to the Caesar Cipher, the key needs to be known by both the sender and the recipient beforehand to securely send messages.

Modern symmetric encryption is very fast and efficient for computers. This makes it the default encryption choice for large files.

Key Sizes

With the advent of high-powered computers, classic cryptography techniques became obsolete. Computers can 'brute force' or 'guess and check' millions of passwords in a second. To resist this, modern symmetric encryption technology - such as the Advanced Encryption Standard (AES) - use very large keys that are hard to guess. These keys can be up to 256-bit! That's 2^{256} or 1.1579×10^{77} possible combinations!

Even with many supercomputers (each of which can try a quadrillion keys a second), it would still take trillions of years to break a single message.

Asymmetric Encryption

Asymmetric encryption (or public-key encryption) is the term for encryption that uses two different, mathematically-linked, keys for encryption and decryption. The 'public key' can be shared freely with anyone and is used to encrypt messages. The other key is called the 'private key' and is not shared with anyone. This key is used to decrypt messages that have been encrypted with the public key.

Visual examples and explanation of asymmetric encryption: https://www.youtube.com/watch?v=AQDCe585Lnc

Modular Arithmetic

Modular arithmetic (also known as a remainder calculation) is a system of arithmetic where numbers "wrap around" when they reach a specific value.

An example of modular arithmetic is calculating time on a 12-hour clock. Classical arithmetic dictates that 9 o'clock + 5 hours would be 14 o'clock. There are only 12 hours on the clock, so the hours "wrap around" leaving you with 2 o'clock instead.

Modular arithmetic is used heavily in asymmetric cryptography.

Calculating Modular Arithmetic

Using modular arithmetic only requires division. When dividing a number by the *modulus* the remainder becomes the solution. This operation is named *modulo*. For example, using the clock analogy from before:

```
9 + 5 (mod 12) =

14 (mod 12) =

14 / 12 = 1 with a remainder of 2.
```

The remainder is the solution, so 2 is your answer.

Calculating Modular Congruence

When two numbers divided by the same modulus have the same remainder, they are considered to be congruent. For example:

```
14 (mod 6) = 2

32 (mod 6) = 2

14 ≡ 32 (mod 6)
```

In this example, 14 and 32 both modulo 6 equal 2. So 14 and 32 are considered to be congruent mod 6.

RSA Algorithm

The most famous and widely-used asymmetric encryption algorithm is RSA. RSA was created in 1977 at MIT University by Ron Rivest, Adi Shamir, and Leonard Adleman. RSA is used to securely send messages on the internet. It protects bank transactions, healthcare information, government messages, and even social media posts or Google searches.

RSA uses modular arithmetic to mathematically link the public and private keys. The public key contains a modulus and a public exponent. The private key contains a private exponent. In the next example, we will see how public and private keys work.

RSA Example

Imagine Alice wants to send a secret message to Bob, but they have no way of meeting in person. They decide to send encrypted messages via the mail. With no way to securely distribute a key for symmetric encryption, they decide to use RSA - an asymmetric encryption algorithm. Alice wants to send Bob the phrase "150" privately:

1. Bob generates the following public key:
    a. Modulus: $n = 3233$
    b. Public Exponent: $e = 17$
2. Bob also generates the following private key:
    a. Private Exponent: $d = 413$
3. Bob sends the public key (n, e) to Alice.
4. Alice encrypts her message using Bob's public key
    a. Alice's plaintext message: $m = 150$
    b. Alice computes ciphertext using formula: $c = m^e \bmod n$.
        i. $c = 150^{17} \bmod 3233 = \mathbf{1685}$
5. Alice sends the ciphertext (c) to Bob
6. Bob decrypts the ciphertext to reveal the secret message.
    a. Bob computes the plaintext (m) using formula: $m = c^d \bmod n$
        i. $m = 1685^{413} \bmod 3233 = \mathbf{150}$
7. Bob can now read the secret message: "150". Because Bob never distributed his private key (d) and Alice never distributes her plaintext message (m), only Bob can decrypt the message.

Transport Layer Security

Transport Layer Security (TLS), formerly Secure Sockets Layer (SSL), is a protocol used to secure internet transactions. It is made up of a combination of modern cryptography techniques, including symmetric encryption and asymmetric encryption.

Everytime you see the green lock or the letters "https://" on your browser, you are using TLS to secure your browsing habits and keep your data secure from prying eyes.