ALICE, BOB, AND CRYPTOGRAPHY

OM MAHESH

Abstract. Two paranoid neighbors, Alice and Bob, use their cryptographic skills to figure out the most secure way to communicate through their bedroom windows.

Introduction

Alice and Bob are neighbors. They hope to meet in secret, so, untrusting of technology, they communicate through their bedroom windows (as shown in Figure 1).

The next day, Alice and Bob meet at the park. However, at the park, they talk about how their communication is still insecure, since anyone passing by can see their messages to one another. So, the next day, they try a new method.

1. Caesar Cipher

(1) Alice constructs a *key* by writing down the alphabet twice, as shown, but the second time "cycling" the letters forward by some integer n.

a b c d e f g h i j k l m n o p q r s t u v w x y z
h i j k l m n o p q r s t u v w x y z a b c d e f g

(2) Alice encrypts her message letter-by-letter using this key (for example, "abcd" becomes "hijk").

(3) Alice writes the message, along with the integer n, to Bob.

(4) Bob, knowing how the Caesar cipher works, uses n to reconstruct the key and decrypt Alice's message.

(5) Bob repeats steps 1 through 4 to send a message back to Alice.

Date: December 12, 2023.

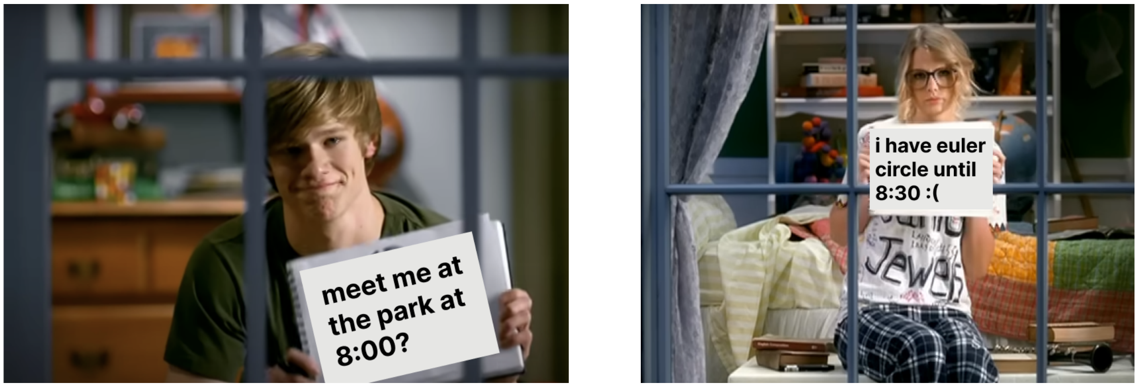
meet me at the park at 8:00?
i have euler circle until 8:30 :(

Figure 1. Alice and Bob planning to meet at the park.

With the Caesar Cipher, Alice and Bob are able to coordinate a time to meet at the park once again. However, at the park, they point out how anyone with a little time on their hands can decrypt their messages by running through the 26 possible values of n. So, the next day, they try a new method.

2. Enigma Machine

1. Alice pulls out her handy "Kenngruppenheft," a German WWII codebook containing specific Enigma machine configurations for each day of the year.
2. After configuring the machine, Alice types a key, producing an electrical signal.
3. The signal first travels into the *plugboard*, a collection of wire connections set to swap 10 pre-determined pairs of letters (the pairs are given by Alice's Kenngruppenheft).
4. The signal then travels through 3 *rotors* (the rotors and their order are given by the Kenngruppenheft). Each rotor applies a permutation to letters passing through it. Then, each time a key is pressed, one (or more) of the rotors rotate, shifting their individual permutations forward. This allows a letter to be encrypted as different letters throughout the message.
5. Next, the signal travels through the *reflector*. The reflector first acts as a plugboard, swapping 13 pre-determined pairs of letters (also given by the Kenngruppenheft). Then, it reflects the signal, sending it backwards through the 3 rotors and the plugboard, repeating steps 3 and 4 in reverse.
6. The signal finally reaches the *lamp panel*, where it lights up a lamp, indicating the encrypted letter to Alice.
7. Alice repeats steps 2 through 6 for each letter of the message.
8. Alice writes the complete encrypted message to Bob.
9. Bob repeats steps 1 through 7, decrypting Alice's message with the same process Alice used to encrypt it (this works because of the reflector, which makes the encryption *symmetric*).
10. Bob repeats steps 1 through 9 to send an encrypted message to Alice.

With their Enigma machines, Alice and Bob are able to coordinate a time to meet at the park once again. However, at the park, they point out the issues with the Enigma machine:

- No letter can be encrypted as itself, due to the way the reflector is configured. This allows attackers to learn information about Alice and Bob's messages.
- If anyone else has the same copy of the Kenngrupenheft and their own Enigma machine, they can easily decrypt the messages.

So, the next day, they try a new method.

3. Diffie-Hellman Key Exchange

1. Alice first chooses her favorite prime p and her favorite primitive root m modulo p. She sends these two numbers to Bob through their bedroom windows, allowing Mallory to see them.
2. Alice then randomly chooses a secret integer a and computes $A = g^a \bmod p$. Meanwhile, Bob secretly chooses b and computes $B = g^b \bmod p$.
3. Alice and Bob publicly exchange A and B, but make sure to keep a and b secret.
4. Alice computes $B^a \bmod p$ while Bob computes $A^b \bmod p$. These are both equal to $k = g^{ab} \bmod p$. This is now their *shared secret key.*

(5) Alice converts her message m to binary m_2 with ASCII ("a" = 01100001, "b" = 01100010, etc.).

(6) Alice converts the shared secret key k to binary k_2.

(7) Alice duplicates k_2 until it is the same length as m_2.

(8) Alice calculates $e_2 = k_2 \oplus m_2$, her final encrypted message. Alice writes e_2 to Bob.

(9) Bob uses k to construct the repeated binary key k_2 in the same way.

(10) Bob decrypts Alice's message by calculating $m_2 = k_2 \oplus e_2$.

(11) Bob repeats steps 5 through 9 to send an encrypted message to Alice.

With the Diffie-Hellman Key Exchange, Alice and Bob are able to coordinate a time to meet at the park once again. However, at the park, they point out the issues with this approach:

- If the number p is "smooth" (i.e. does not have any large prime factors), the Pohlig-Hellman algorithm [1] can be used to find their shared secret key.
- Repeating and reusing the key allows attackers to learn information about their messages. For example, $e_2 \oplus E_2 = m_2 \oplus M_2$ for two messages m_2 and M_2 encrypted as e_2 and E_2, respectively, with the same shared secret key.

So, the next day, they try a new method.

4. Elliptic Curve Diffie-Hellman

(1) Alice first picks one of her favorite elliptic curves and primes p. Usually, they are one of the following pairs.

$$y^2 = x^3 - 3x + 2455155546008943817740293915197451784769108058161191238065$$
$$p = 6277101735386680763835789423207666416083908700390324961279$$

$$y^2 = x^3 - 3x + 18958286285566608000408668544493926415504680968679321075787234672564$$
$$p = 26959946667150639794667015087019630673557916260026308143510066298881$$

$$y^2 = x^3 - 3x + 41058363725152142129326129780047268409114441015993725554835256314039467401291$$
$$p = 115792089210356248762697446949407573530086143415290314195533631308867097853951$$

(2) Alice picks a point $G = (x, y)$ of order n on the curve such that $n - 1$ is not smooth (i.e. has a large prime factor).

(3) Alice sends her elliptic curve E, the prime p, the point G, and its order n to Bob.

(4) Alice chooses random integer a and calculates $A = aG$ while Bob chooses random integer b and calculates $B = bG$.

(5) Alice and Bob publicly exchange A and B, but keep a and b secret.

(6) Alice and Bob independently calculate their shared secret key $k = bA = aB$.

(7) Alice converts her message m to ASCII binary m_2.

(8) Alice truncates m_2 to the size of k_2 and calculates $e_2 = m_2 \oplus k_2$, her encrypted message. Alice writes e_2 to Bob.

(9) Bob gets e_2 and k_2 from $e_1 6$ and k, and then calculates Alice's message $m_2 = e_2 \oplus k_2$.

(10) Alice and Bob repeat steps 1 through 9 until the rest of Alice's message is sent to Bob.

(11) Alice and Bob repeat steps 1 through 10 for Bob to send a message to Alice.

With the Elliptic Curve Diffie-Hellman approach, Alice and Bob are able to coordinate a time to meet at the park once again. This time, Alice and Bob agree that they are satisfied with their encryption algorithm.

Until finally Alice says, “Now. . . how do we communicate with our other neighbor, Carol?”

References

[1] S. C. Pohlig and M. E. Hellman, “An improved algorithm for computing logarithms over gf(p) and its cryptographic significance.” https://www-ee.stanford.edu/~hellman/publications/28.pdf, 1978.