Cryptography Project

Part 1: Secure Chat

Description

Chat apps bring people closer virtually even if they live far off from each other. It gives them the convenience to connect without the need to spend a fortune and time to meet people in-person. However, ensuring the security of messages is a huge challenge that is critical for the end users. Here is where things become interesting and cryptography kicks in.

Each encryption algorithm has its strengths and weaknesses. For instance, AES is a symmetric algorithm that is fast enough and efficient for large data, but the encryption key must be shared securely, so we tend to use Deffie Hellman for key exchange. However, DH is prone to man-in-the-middle attacks. To make things secure, a digital signature is required for making sure that only the source is the one who sent this message.

You are required to build a secure chatting application over sockets that uses Deffie Hellman for exchanging session keys, Elgamal Digital Signature for signing key exchange messages and AES for actual messages encryption.

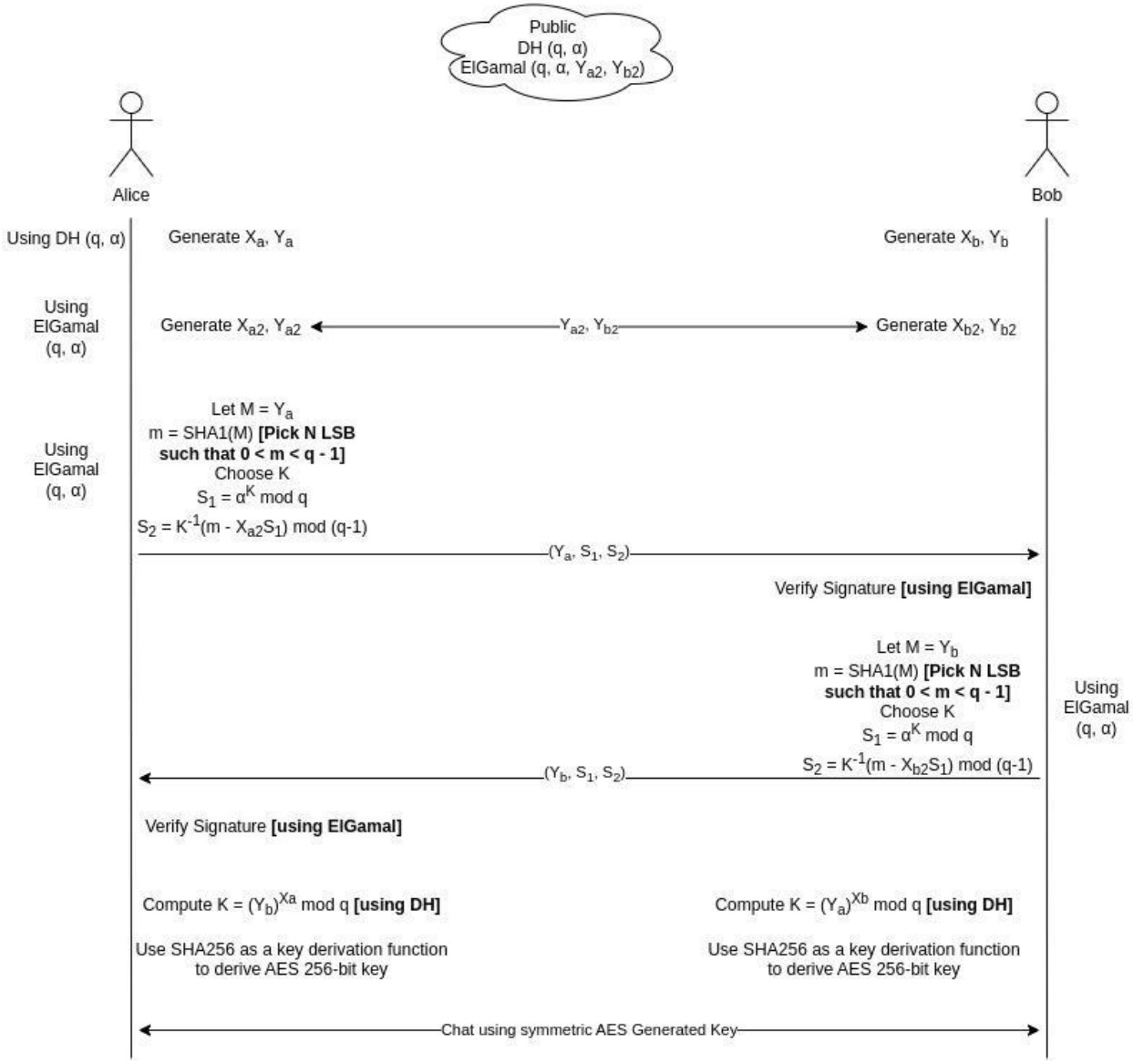
Check Key Exchange Problems - Computerphile for clarification.

You need to implement all algorithms from scratch except for AES and the hash function (use SHA1) for Elgamal Digital Signature.

Establishing Connection

- Deffie Hellman keys (q, α, Xa, Ya, Xb, Yb) are different from ElGamal keys (q, α. Xa2, Ya2, Xb2, Yb2)
- Elgamal keys should be long-living (generated once a year) while DH keys are somehow Ephemeral (generated per session) - *won't differ in our implementation as both keys are always generated once the program runs*
- Both parties will read DH(q, α) and Elgamal (q, α) from a file simulating being publicly accessed.
- Both parties will generate two public/private-key pairs (one for DH and one for Elgamal)
- Alice and Bob will exchange Elgamal keys.
- Alice sends to Bob her DH public key after signing it using Elgamal digital signature
- Bob verifies Alice's identity and terminates the connection if the signature is not valid.
- Bob sends back his DH public key after signing it using Elgamal digital signature
    - Feel free to import a ready made function for SHA1 hash function
- Alice verifies Bob's identity and terminates the connection if the signature is not valid.

- Both parties compute the DH shared secret
- Generate an AES 256-bit key from the DH shared key using a key derivation function SHA256(shared secret)
    - Feel free to import a ready made function for SHA256 hash function
- Use the generated key for the subsequent chat messages between the two parties for encryption and decryption.
    - Feel free to import a ready made implementation for AES



Deliverables

- A program that does encryption/decryption.
    - Should be able to run two instances of this program and start chatting.
    - Take care of what should be public and what should be private.
- README file explaining how to run your application.

Part 2: CTF

In this part you will solve 8 CTF tasks and you can find them attached with the document.
Each task will ask you to find a hidden flag in a different way and your job to find this flag correctly.
You may deal with images, audio, network, or web pages.

The flag structure can be one of three:

1. CMPN{some_text}
2. picoCTF{some_text}
3. fastctf{some_text}
4. SOME_TEXT

CTF – 1 (Cryptanalysis)

You are provided with a text file “encrypted_text.txt”, inside it there’s an extremely long ciphered text, can you figure out what’s the original text?

CTF – 2 (Packet Analysis)

You have been challenged by a friend to find a flag in their network packets file “packets.pcapng”, they have left the flag in there and have manipulated it to not give you an easy time, can you find out what the original text is from the packet data ?

CTF – 3 (Image Manipulation)

You’re going through an old hard drive, you come across these 2 images “first.png” & “second.png”, can you make a flag out of them?

CTF – 4 (Bit Shifting)

Find the flag in the text file “bits.txt” (Hint: When in doubt, pause and shift)

CTF – 5 (Search)

There’s a flag in the file “logs” (Hint: grep)

CTF – 6 (New Encryption)

The attackers have developed an unbreakable encryption, can you break it? When you break it decrypt the ciphertext found in cipher.txt

CTF – 7 (Steganography)

Steganography is to hide some file or data inside another file or data. In the given image something is HIDING.

CTF – 8 (Can You Help Me ?)

The terrorists have caught someone in danger, can you find out where they are to help them?

Deliverables

- A document/markdown file outlining the steps you followed to solve every problem specifying the flag you found.
- Any code used to solve any of the CTF problems should be included in your submission
- CTF Problem 6 specifically will not be graded without the code!

Rules

- Group size is up to 4 members.
- Submit a zip file containing your team's work for both parts.
- You are free to use any language but make sure you can handle large numbers (hundreds of digits).
- All students should submit their own-written code without the help of any external source. Copied projects from each other or from the internet will not be accepted.

Due Date: 6 May at 11:59 PM

Submit to: https://forms.gle/VuhNJD6HSv8HiW9H8