Cryptography Engineering

- Lecture 9 (Dec 17, 2025)

- Remarks on the homework

Certificate

- Make sure you appropriately use certificates

- Q1. Who issues (signs) the server certificate in TLS?
  - a) The client
  - b) The server
  - c) The certificate authority

Certificate

- Make sure you appropriately use certificates

- Q2. In a typical client-server TLS connection, whose public key is certified (signed) by the CA?
  a) The client
  b) The server
  c) The certificate authority

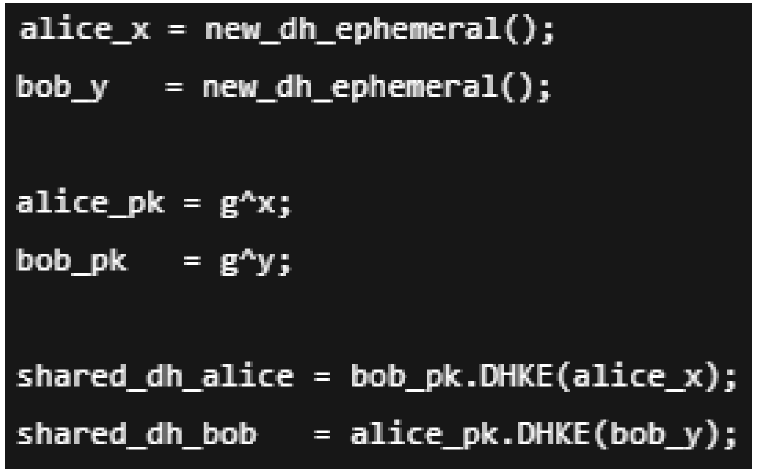
Certificate

- Make sure you appropriately use certificates

- Q3. Which public key should we use to verify the server's certificate?
  - a) The client's public key
  - b) The server's public key
  - c) The CA's public key

Protocol Flow

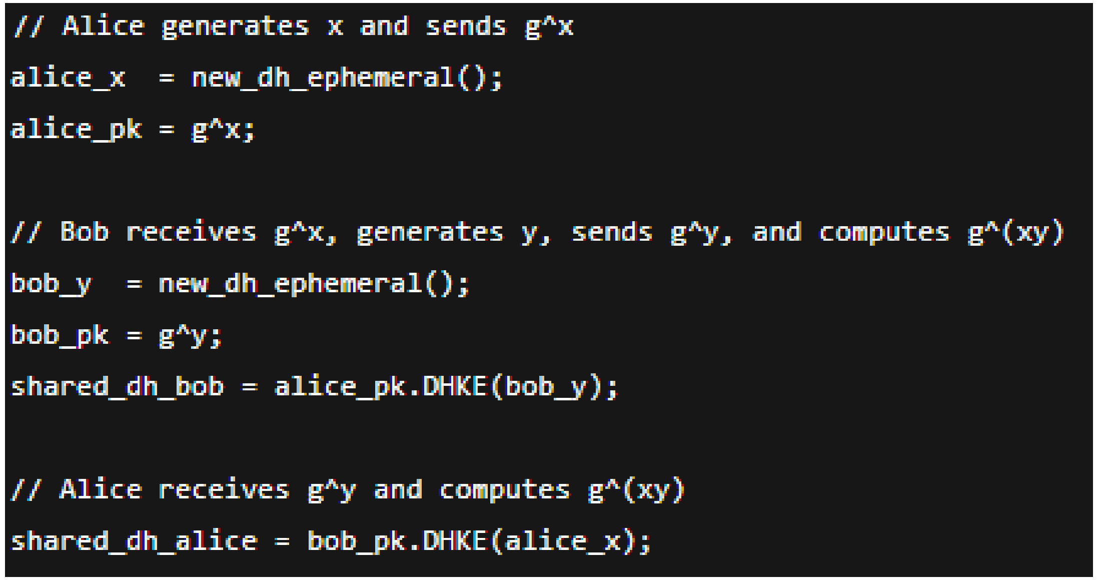
- Please make sure your implementation faithfully follows the protocol flow, even though we are not using sockets.
- What is the difference between the following two versions of Diffie–Hellman key exchange (DHKE)?



```
alice_x = new_dh_ephemeral();
bob_y   = new_dh_ephemeral();

alice_pk = g^x;
bob_pk   = g^y;

shared_dh_alice = bob_pk.DHKE(alice_x);
shared_dh_bob   = alice_pk.DHKE(bob_y);
```



```
// Alice generates x and sends g^x
alice_x  = new_dh_ephemeral();
alice_pk = g^x;

// Bob receives g^x, generates y, sends g^y, and computes g^(xy)
bob_y  = new_dh_ephemeral();
bob_pk = g^y;
shared_dh_bob = alice_pk.DHKE(bob_y);

// Alice receives g^y and computes g^(xy)
shared_dh_alice = bob_pk.DHKE(alice_x);
```

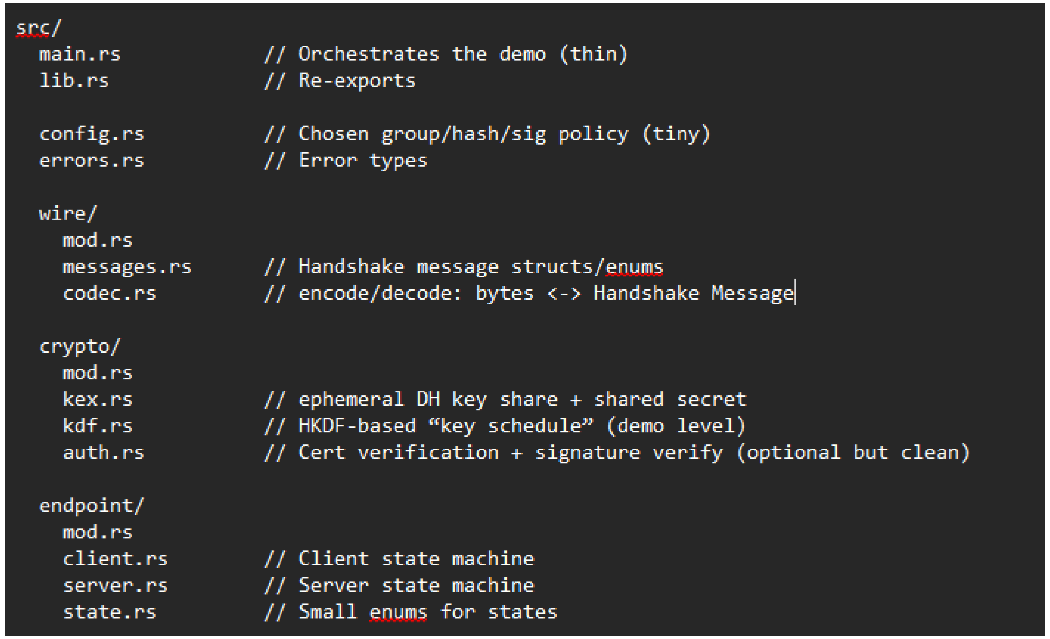
Modular Programming & Separation of Concerns

- Please do **not** put everything into one huge `main.rs`!
  - For homework, this is acceptable.
  - But for the final project, you will **lose points** for this.

Modular Programming & Separation of Concerns

- Each function does one thing.
- Each module groups functions for one *concern* (topic).
- Expose a clean, reusable interface for future code.

- A good example of a TLS demo



```
src/
  main.rs             // Orchestrates the demo (thin)
  lib.rs              // Re-exports

  config.rs           // Chosen group/hash/sig policy (tiny)
  errors.rs           // Error types

  wire/
    mod.rs
    messages.rs       // Handshake message structs/enums
    codec.rs          // encode/decode: bytes <-> Handshake Message

  crypto/
    mod.rs
    kex.rs            // ephemeral DH key share + shared secret
    kdf.rs            // HKDF-based “key schedule” (demo level)
    auth.rs           // Cert verification + signature verify (optional but clean)

  endpoint/
    mod.rs
    client.rs         // Client state machine
    server.rs         // Server state machine
    state.rs          // Small enums for states
```

On the Usage of AI Tools

- Using AI tools is **welcome.**

- **However, you must understand and be able to explain the core of your solution**, including:
  - the overall code structure and logic,
  - the protocol/algorithm you implemented,
  - what each component is for and how they interact/work together,
  - why it achieves the required security/functionality.

- In the final oral exam, questions may be based on:
  - (primarily) Your final project and report
  - Your homework submissions

Have a lovely Christmas break!

Just a quick reminder: Homework 2 is due on ***9 January 2026****.*