**Exercises (week\_1)**

1. **Symmetric vs. Asymmetric Encryption**: What are the key differences between symmetric and asymmetric encryption? Provide a practical use case for each.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Mã hoá đối xứng | Mã hoá bất đối xứng | Lý do |
| Số lượng key | 1 Key share dùng để mã hoá và giải mã | cặp key : public để mã hoá và chỉ có private có thể giải mã |  |
| Tốc độ | Nhanh hơn, nhẹ hơn | Chậm hơn, nặng hơn về tính toán | Do thuật toán và các cơ chế toán học  Độ dài khoá (128 bit, 256 bit vs 2048 bit, 3072 bit,..) |
| Bảo mật chia sẽ khoá | cần phương pháp trao đổi khoá an toàn | Không cần | DX: key share cần giấu kín  BDX: Key public được công khai |
| Ứng dụng | VPN → Dùng AES để mã hóa dữ liệu giữa hai site sau khi trao đổi khóa. | Email bảo mật (PGP) → Người gửi mã hóa bằng khóa công khai của người nhận, người nhận giải mã bằng khóa riêng. |  |

1. **Public-Key Cryptography and Key Exchange Protocols**: How can the Diffie-Hellman protocol enhance security in a messaging application? (Diffie-Hellman và bảo mật ứng dụng nhắn tin)

- Đầu tiên với app nhắn tin cần mã hoá đối xứng vì không tốn nhiều tài nguyên như nói ở trên ( vì truyền nhiều dữ liệu nhỏ : tin nhắn, hình ảnh, file...)

- Diffie Hellman hỗ trợ tốt khi tạo secret key một cách công khai và hiệu quả nhờ bài toán khó trong việc “khó tính logarit rời rạc trong trường số nguyên modulo p"

Ví dụ với alice và bob, công khai trao đổi một số giá trị được tính toán, nhưng bản thân những giá trị này không tiết lộ khóa riêng của họ hoặc khóa bí mật cuối cùng. Kẻ gian có thể tìm ra nhưng rất khó.

- Khi khóa bí mật chung được thiết lập, nó có thể được sử dụng làm khóa đối xứng cho một thuật toán mã hóa nhanh và hiệu quả (như AES) để mã hóa tất cả các tin nhắn tiếp theo.

1. **Hash Functions**: What features make SHA-256 and Poseidon good hash functions for ensuring data integrity? Mention one unique advantage of Poseidon.

- Tính chất chung đảm bảo toàn vẹn dữ liệu:

- Không thể đảo ngược (Preimage resistance)

- Chống va chạm (Collision resistance)

- Đầu vào nhỏ thay đổi → đầu ra thay đổi hoàn toàn (Avalanche effect)

Điểm đặc biệt của Poseidon: Rất tiết kiệm chi phí tính toán trong mạch Zero-Knowledge Proof, phù hợp cho zkRollup, zkSNARK, blockchain Layer 2.

1. **Merkle Trees**: Explain how Merkle trees can help verify data in a large database efficiently.

- Không cần show toàn bộ dữ liệu: Để xác minh một khối dữ liệu cụ thể trong một cơ sở dữ liệu lớn, chỉ cần khối dữ liệu muốn xác minh, giá trị băm của nó (nút lá), và một tập hợp các giá trị băm trên đường đi từ nút lá của dữ liệu bạn muốn xác minh đến Gốc Merkle.

Ví dụ, để xác minh Data3, bạn cần Hash(Data3), Hash(Data4), Hash(Hash(Data1)+Hash(Data2)). Kết hợp tất cả để tạo ra gốc giả định rồi so sánh với gốc được công bố. Nếu chúng khớp, bạn có thể khẳng định rằng Data3 (và toàn bộ dữ liệu phụ thuộc trên đường đi) là hợp lệ và không bị thay đổi.

- Hiệu quả về kích thước: Nếu có N khối dữ liệu lá, bằng chứng Merkle cho một lá cụ thể chỉ bao gồm log2(N) giá trị băm. Điều này là cực kỳ hiệu quả khi N rất lớn (ví dụ: hàng tỷ bản ghi), vì log2 (N) sẽ là một con số rất nhỏ so với N.

1. **Cryptographic Commitments**: How can Pedersen Commitments be used in a blockchain protocol to maintain transaction privacy?

Phương pháp cam kết một giá trị mà:

- Không lộ giá trị cho người ngoài.

- Không thể thay đổi giá trị sau khi cam kết.

Công thức: C = g^m \* h^r (với m là giá trị cần cam kết, r là số ngẫu nhiên bí mật)

Ứng dụng trong Blockchain:

Confidential Transactions (CT):

Ẩn số lượng coin trong giao dịch nhưng vẫn đảm bảo tổng đầu vào = tổng đầu ra (giúp kiểm toán mà không lộ số tiền).

Tính chất chính:

- Hiding: Giấu giá trị.

- Binding: Không thể thay đổi giá trị sau khi cam kết.

1. **Digital Signatures**: How can you verify the authenticity of a digitally signed document?

- Người gửi :

+ **hash** tài liệu sau đó **ký** bằng private key ( chữ ký số )

+ **gửi** cả tài liệu gốc ( plaintext ) và chữ ký số trên

- Người nhận :

+ Nhận tài liệu gốc ( plaintext ) và **hash** lại

+ Nhận chữ ký số, **giải mã** bằng public key người của gửi và so sánh với bản hash trên

- Nếu trùng khớp thì tài liệu không bị sửa đổi, đúng người gửi.