[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy
Scribed by: Kunj Thakkar (202251142)

Autumn 2024-2025
Lecture (Week 11)

Software Stream Cipher

Detailed Explanation

A stream cipher encrypts data bit-by-bit or byte-by-byte using a generated keystream. This keystream depends on a secret key and often an initialization vector (IV), ensuring the encryption is unique per message or session.

Encryption Process

Key stream generation is done with a pseudorandom number generator (PRNG) using the key and IV. Each bit of plaintext is XORed with the keystream to create the ciphertext.

Example - Simplified Stream Cipher

Suppose the plaintext `HELLO` is encrypted using ASCII values with a pseudorandom keystream. The ASCII values for the letters are $H = 72, E = 69, L = 76, L = 76, O = 79$. A random keystream could be $20, 45, 30, 50, 15$. XORing each plaintext value with the corresponding keystream gives ciphertext values $92, 104, 110, 126, 94$.

Decryption

To decrypt, XOR the ciphertext back with the same keystream to recover the plaintext.

RC4 Stream Cipher

RC4, developed by Ron Rivest in 1987, is a symmetric stream cipher that became popular due to its simplicity and speed. The name "RC" stands for "Rivest Cipher." Despite its widespread use, RC4 is not preferred anymore due to vulnerabilities.

Algorithm

The algorithm involves three main steps. First, the **Key Initialization** step expands a variable-length key into an initial state permutation array using the Key Scheduling Algorithm (KSA). Second, the **Pseudo-Random Bit Generation** step produces the keystream. Finally, in the encryption or decryption phase, the keystream is XORed with plaintext or ciphertext, respectively.

Drawbacks

RC4 has various vulnerabilities, including biases in its output and key scheduling weaknesses. While still in use for some legacy systems, modern cryptography standards like AES are preferred.

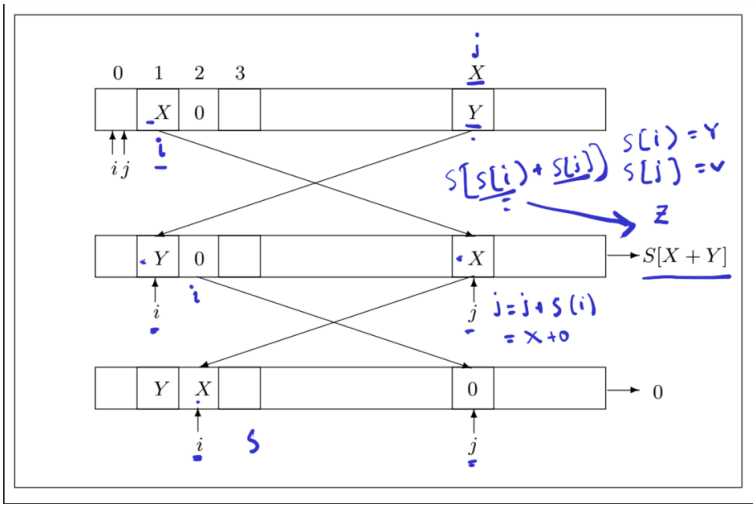

Figure 1: The Biased Second Output of RC4

RC4 PRGA

Detailed Explanation

RC4 uses two algorithms. The KSA initializes a permutation array S with 256 bytes based on a secret key. The PRGA generates the keystream by swapping array values and deriving output bytes.

Example of RC4

For example, to encrypt $M = 77$ using $K = [1, 2, 3]$, the KSA initializes S and permutes it based on K. During PRGA, i and j indices iterate over S, producing a keystream value $K = 95$. XORing $M = 77$ with $K = 95$ gives $C = 18$. Decrypting $C = 18$ with the same $K = 95$ recovers $M = 77$.

Zero-Knowledge Proof Using DLP

Detailed Explanation

Zero-Knowledge Proofs allow a prover to demonstrate knowledge of a secret without disclosing it.

Example - Using Discrete Logarithm Problem (DLP)

Prover P knows x in $h = g^x \mod p$. Public values are $p = 23, g = 5, h = 8$, with $x = 3$.

Commitment

P picks $r = 4$, computes $t = g^r \mod p = 4$, and sends t to V.

Challenge and Response

V sends $c = 1$, and P calculates $s = r + cx \mod (p - 1) = 7$ and sends s.

Verification

V verifies $g^s \mod p = t \cdot h^c \mod p$, confirming the proof.

Detailed Components of the Signal Protocol

Key Components

Signal uses the Identity Key (IK), Signed Prekey (SPK), One-Time Prekeys (OTPK), Ephemeral Keys (EK), Master Secret (SK), and Session Keys.

1. Signal Protocol in Two-Party Communication

A. Session Setup: X3DH

In the X3DH protocol, Alice retrieves Bob's IK_B, SPK_B, and $OTPK_B$ keys and computes a shared secret SK using Diffie-Hellman operations.

B. Double Ratchet Algorithm

Alice and Bob continually update keys after every message for forward secrecy. Keys are derived from a chain key and discarded after use.

2. Signal Protocol in Group Chats

A. Sender Key Distribution

Each participant creates a Sender Key comprising a Message Key and Signing Key. These are shared via X3DH.

B. Messaging in Group Chats

Messages are encrypted with the Sender Message Key and signed with the Sender Signing Key. Recipients verify signatures and decrypt messages.

3. Security Features of the Signal Protocol

A. Forward Secrecy and Post-Compromise Security

The Double Ratchet ensures unused keys are discarded for forward secrecy. Compromised keys don't affect future messages.

B. Deniability

Messages are not cryptographically attributable to the sender.

Example Walkthrough: Alice and Bob's Conversation

Setup

Alice and Bob exchange public keys to derive a shared secret SK.

Alice Sends a Message

Alice derives a message key MK_A from her chain key and encrypts the message "Hello, Bob!" into ciphertext C.

Bob Receives and Decrypts

Bob updates his chain key, derives MK_A, and decrypts C to get "Hello, Bob!" Discarded keys ensure forward secrecy.