



Cybersecurity

Professor: F. d'Amore

Table of Contents

1 Introduction	5
2 Symmetric Encryption	6
2.1 Stream Ciphers	7
2.2 Types of Stream Ciphers	8
2.3 Stream Ciphers in practice	8
2.3.1 A5/1	9
2.3.2 Rivest Cipher (RC4)	9
2.3.3 Salsa20	9
2.4 Block Ciphers	10
2.5 Block Ciphers in practice	10
2.5.1 Data Encryption Standard (DES)	10
2.5.2 Advanced Encryption Standard (AES)	11
2.5.3 Twofish	12
3 Data Integrity	13
4 Public Key Cryptography	14
5 Digital Signatures	15
6 Cryptographically Secure Pseudo-Random Number Generators	16
7 Authentication	17
8 Secret Sharing	18
9 Access Control	19
10 Secure Protocols	20
11 Firewalls	21
12 Email Security	22
13 Web Technologies	23
14 Web Security	24
15 Web Tracking	25

List of Figures

1	Basic Structure of DE	11
2	Basic Structure of AE	12

List of Tables

1 Introduction

2 Symmetric Encryption

Secret key cryptography involves the use of a single key. Given a message (the plaintext) and the key, encryption produces unintelligible data which is about the same length as the plaintext was.

Secret key cryptography is sometimes referred to as **conventional cryptography** or **symmetric cryptography**.

Secret key encryption schemes require that both the party that does the encryption and the party that does the decryption share a secret key. We will discuss two types of secret key encryption schemes:

- **Stream Ciphers:** This uses the key as a seed for a pseudorandom number generator, produces a stream of pseudorandom bits, and \oplus s (bitwise exclusive ors) that stream with the data. Since \oplus is its own inverse, the same computation performs both encryption and decryption.
- **Block Ciphers:** This takes as input a secret key and a plaintext block of fixed size (older ciphers used 64-bit blocks, modern ciphers use 128-bit blocks). It produces a ciphertext block the same size as the plaintext block. To encrypt messages larger than the blocksize, the block cipher is used iteratively with algorithms called *modes of operation*. A block cipher also has a decryption operation that does the reverse computation.

So, block ciphers encrypt data in blocks of set lengths, while stream ciphers do not and instead encrypt plaintext one byte at a time. The two encryption approaches, therefore, vary widely in implementation and use cases.

2.1 Stream Ciphers

Idea: try to simulate one-time pad

A stream cipher encrypts a continuous string of binary digits by applying time-varying transformations on plaintext data. Therefore, this type of encryption works bit-by-bit, using keystreams to generate ciphertext for arbitrary lengths of plain text messages. The cipher combines a key (128/256 bits) and a nonce digit (64-128 bits) to produce the **keystream** — a pseudorandom number XORed with the plaintext to produce ciphertext. While the key and the nonce can be reused, the keystream has to be unique for each encryption iteration to ensure security. Stream encryption ciphers achieve this using feedback shift registers to generate a **unique nonce (number used only once) to create the keystream**.

Encryption schemes that use stream ciphers are less likely to propagate system-wide errors since an error in the translation of one bit does not typically affect the entire plaintext block. Stream encryption also occurs in a **linear, continuous manner**, making it simpler and faster to implement. On the other hand, stream ciphers lack diffusion since each plaintext digit is mapped to one ciphertext output. Additionally, they do not validate authenticity, making them vulnerable to insertions. If hackers break the encryption algorithm, they can insert or modify the encrypted message without detection. Stream ciphers are mainly used to encrypt data in applications where the amount of plain text cannot be determined and in low latency use-cases.

2.2 Types of Stream Ciphers

Stream ciphers fall into two categories:

Synchronous Stream Ciphers:

- In a synchronous stream cipher, the keystream block is generated independently of the previous ciphertext and plaintext messages. This means that **each keystream block is generated based only on the key** and does not depend on any previous blocks.
- The most common stream cipher modes use pseudorandom number generators (PRNGs) to create a string of bits, which is combined with the key to form the keystream.
- The keystream is then XORed with the plaintext to generate the ciphertext.

Self-Synchronizing/Asynchronous Stream Ciphers:

- A self-synchronizing stream cipher, also known as ciphertext autokey, **generates the keystream block as a function of both the symmetric key and the fixed-size (N-bits) previous ciphertext block**.
- By altering the ciphertext, the content of the next keystream is changed. This property allows self-synchronizing ciphers to detect active attacks because any modification to the ciphertext will affect the decryption of subsequent blocks, making it easier to detect tampering.
- Asynchronous stream ciphers also provide limited error propagation. If there is a single-digit error in the ciphertext, it can affect at most N bits (the size of the previous ciphertext block) in the next keystream block

.

2.3 Stream Ciphers in practice

popular encryption schemes that use stream ciphers include:

2.3.1 A5/1

2.3.2 Rivest Cipher (RC4)

2.3.3 Salsa20

2.4 Block Ciphers

Block ciphers convert data in plaintext into ciphertext in fixed-size blocks. The block size generally depends on the encryption scheme and is usually in octaves (64-bit or 128-bit blocks). If the plaintext length is not a multiple of 8, the encryption scheme uses **padding** to ensure complete blocks. For instance, to perform 128-bit encryption on a 150-bit plaintext, the encryption scheme provides two blocks, 1 with 128 bits and one with the 22 bits left. 106 Redundant bits are added to the last block to make the entire block equal to the encryption scheme's ciphertext block size.

While Block ciphers use symmetric keys and algorithms to perform data encryption and decryption, they also require an **initialization vector (IV)** to function. An initialization vector is a pseudorandom or random sequence of characters used to encrypt the first block of characters in the plaintext block. The resultant ciphertext for the first block of characters acts as the initialization vector for the subsequent blocks. Therefore, the symmetric cipher produces a unique ciphertext block for each iteration while the IV is transmitted along with the symmetric key and does not require encryption.

Block encryption algorithms offer **high diffusion**; that is, if a single plaintext block were subjected to multiple encryption iterations, it resulted in a unique ciphertext block for each iteration. This makes the encryption scheme relatively tamper-proof since it is difficult for malicious actors to insert symbols into a data block without detection. On the other hand, block ciphers have a **high error propagation rate** since a bit of change in the original plaintext results in entirely different ciphertext blocks.

2.5 Block Ciphers in practice

2.5.1 Data Encryption Standard (DES)

DES is a symmetric block cipher using 64 bit blocks and 56 bits key.

The choice of using 56 bits for keys in DES was a compromise between security

and practicality. While a longer key would provide stronger security, the designers of DES also needed to consider the limitations of computing technology at the time. The inclusion of 8 parity bits reduced the effective key size to 56 bits, which some experts even then considered inadequate for robust security. However, the decision to use a 56-bit key was likely influenced by a balance between security requirements and the feasibility of implementing and processing longer keys with the available hardware during that era.

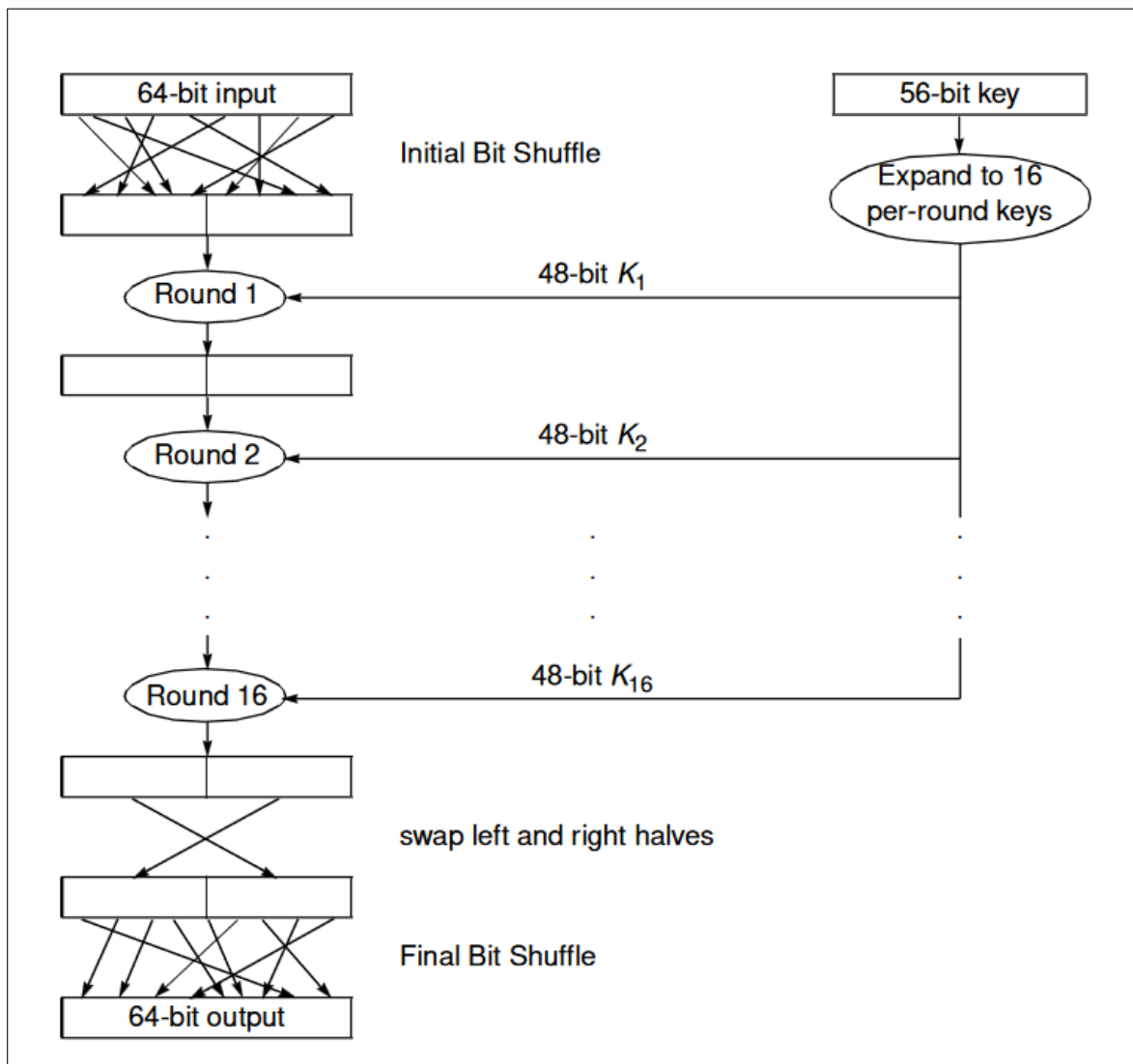


Figure 1: Basic Structure of DE

2.5.2 Advanced Encryption Standard (AES)

AES is a symmetric block cipher using 128 bit blocks and 128, 192 or 256 bits key with the resulting variants called AES-128, AES-192, and AES-256.

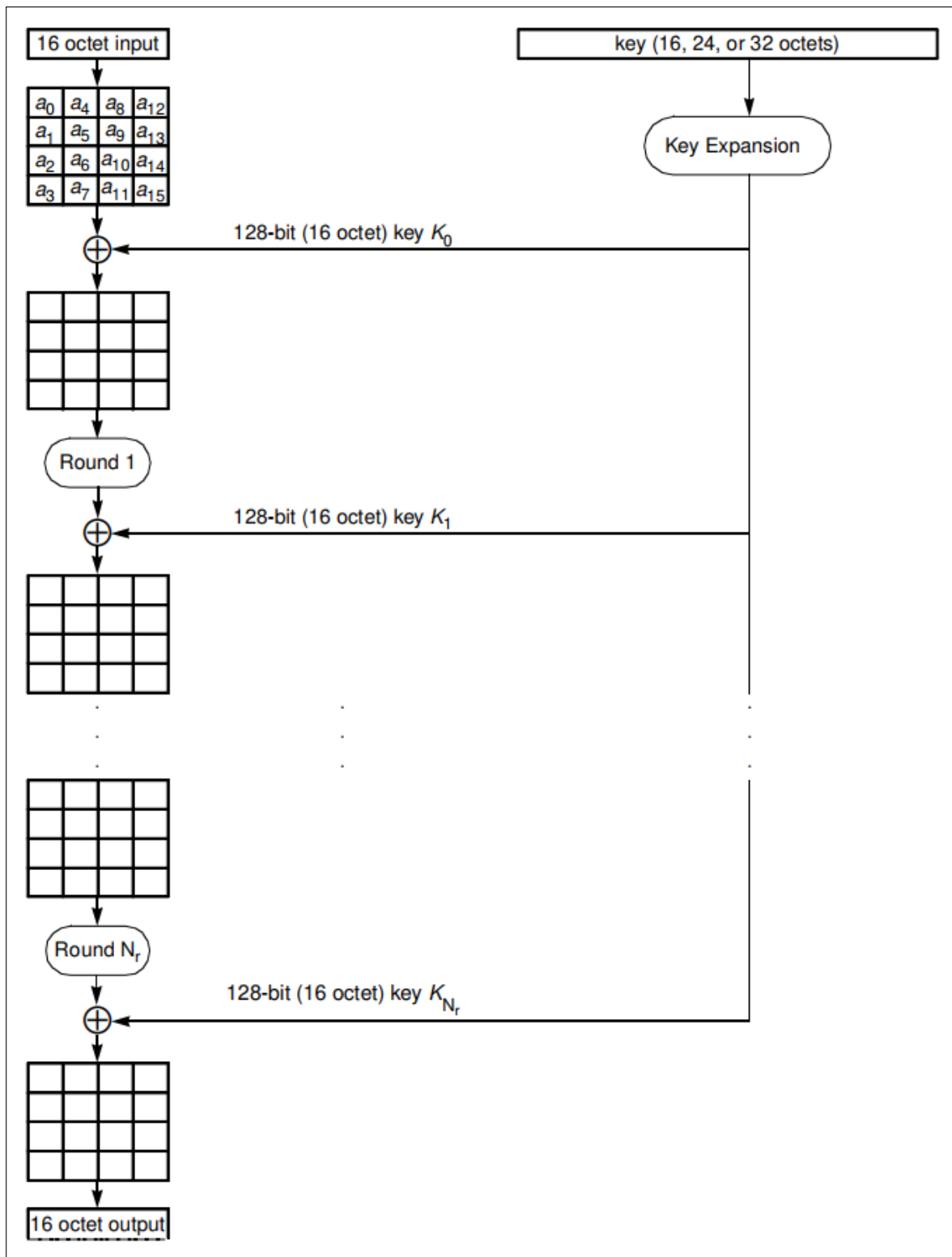


Figure 2: Basic Structure of AE

2.5.3 Twofish

3 Data Integrity

4 Public Key Cryptography

5 Digital Signatures

6 Cryptographically Secure Pseudo-Random Number Generators

7 Authentication

8 Secret Sharing

9 Access Control

10 Secure Protocols

11 Firewalls

12 Email Security

13 Web Technologies

14 Web Security

15 Web Tracking