

Feedback based Modes

- ▶ Uses an **affine function** that takes a **block cipher output** and a **plain text** block to produce the corresponding cipher text block and an updated state which is used as the next block cipher input
- ▶ **Reduce** the **state memory**, at the cost of losing parallelizability

Target Applications

- ▶ This is one of the most popular method of constructing **area-efficient** block cipher based AE.
- ▶ Primarily targeted for **resource constrained environments** such as RFID tags, sensor networks.
- ▶ Typically **inverse-free** and area-efficient for combined enc-dec implementations.

Types of Feedback

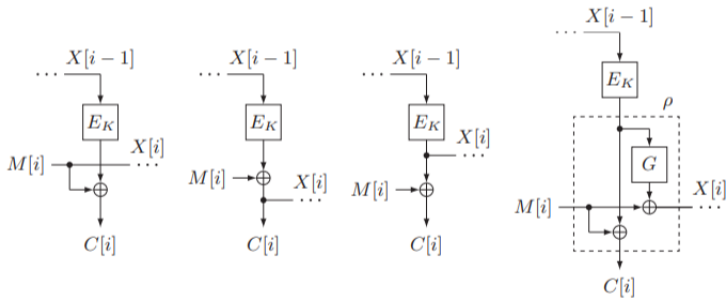


Figure: Hybrid Feedback functions: (a) PFB, (b) CFB, (c) OFB, (d) CoFB

Types of Hybrid Feedback

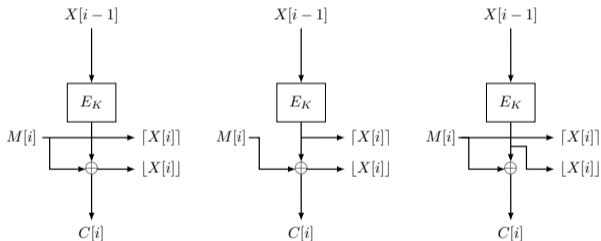


Figure: (a) PFB+CFB, (b) OFB+CFB, (c) OFB+PFB

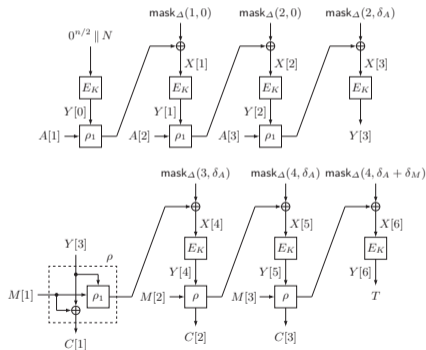
Security of Rate-1 Feedback based AE

Encryption	Decryption	Additional states to achieve Security
PFB	CFB	n -bits
CFB	PFB	-
OFB	OFB	-
CoFB	CoFB	$n/2$ -bits
HyFB (CFB+PFB)	HyFB (PFB+CFB)	$n/2$ -bits
HyFB (CFB+OFB)	HyFB (PFB+OFB)	-
HyFB (PFB+OFB)	HyFB (CFB+OFB)	-

From Combined to Hybrid: Making Feedback-based AE Even Smaller
[Chakraborti et al., ToSC 2020]

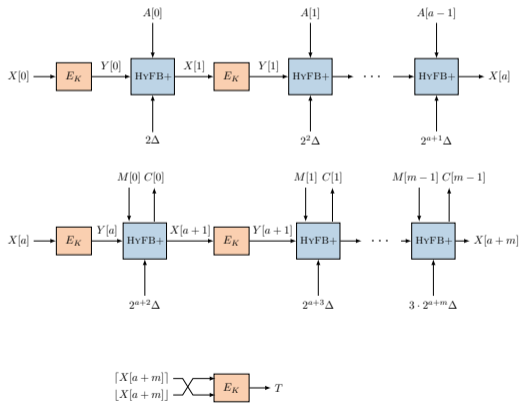
For any rate-1 feedback-based AE with additional state of τ -bits, there is an adversary that breaks the construction with query complexity 2^τ

Example1: COFB (A Mode with Combined Feedback)



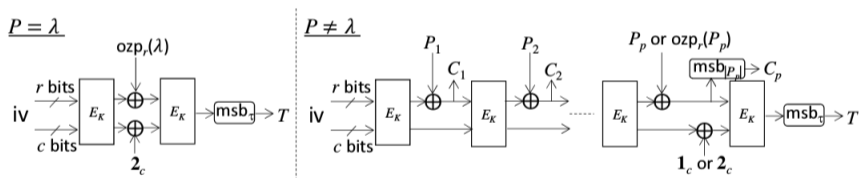
► Inverse-free, Rate-1, State size: $1.5n$ -bit (optimal for rate-1)

Example2: HyENA (A Mode with Hybrid Feedback)



► Inverse-free, Rate-1, State size: $1.5n$ -bit (optimal for rate-1)

Example3: SAEAES (A Block-cipher based Sponge Variant)



- ▶ Inverse-free, Rate-1/2, State size: only n -bit (at the cost of reduced rate)

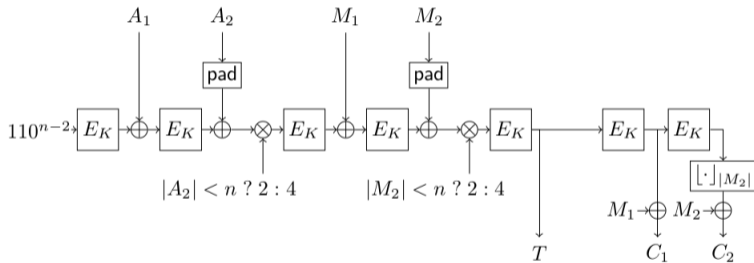
SIV based Modes

- ▶ Deterministic Authenticated Encryption.
- ▶ Follows **MAC-then-Encrypt** structure, and hence **two pass** mode
- ▶ Typically obtain **single-state** implementation

Target Applications

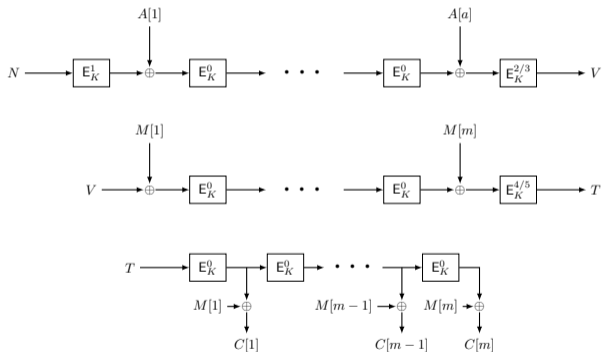
- ▶ Provides **defense in depth**, i.e., maximal robustness even in undesired environment such as when nonces are repeated.
- ▶ Supports **efficient short input** data processing, while minimizing the memory consumption and precomputation. In use cases with tight requirements on delay and latency, the typical packet sizes are small (way less than 1 KB)
- ▶ Excellent choice for **energy efficient** designs, used in devices that operate on a tight energy budget such as handheld devices, medical implants or RFID tags.

Example1: SUNDAE



- ▶ Use **CBC MAC** with **OFB** encryption
- ▶ **Single-state, Inverse-free, efficient for short messages**

Example2: ESTATE



- ▶ Single-state, Inverse-free, efficient for short messages
- ▶ Use tBC to ensure RUP security, optimal block-cipher invocations

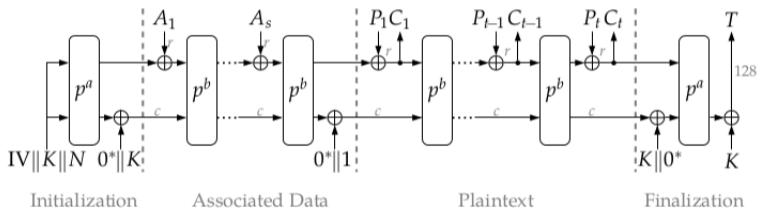
Sponge Modes

- ▶ Use **public permutation** instead of keyed permutation
- ▶ Employs **duplex** mode of operation - absorbs the data and then squeeze the ciphertext
- ▶ Has the advantage of key agility: no key-scheduling

Target Application

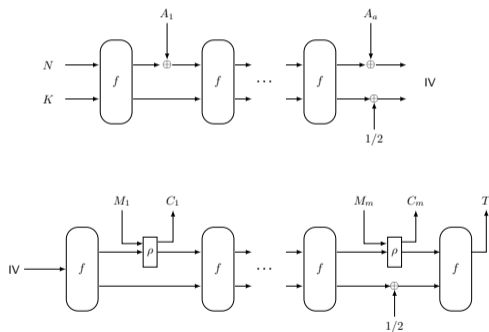
- ▶ Well suited for achieving a balance between **hardware cost** and **software efficiency**.
- ▶ **Versatile** in nature and can be tuned to achieve good performance in any domain, including **high speed implementation**, **memory-restricted environment** and usual desktop computers.

Example1: ASCON (Simple Duplex type Sponge Mode)



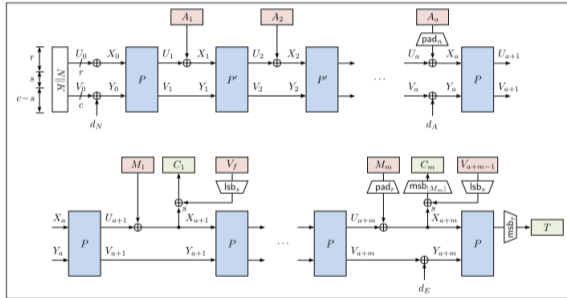
- ▶ Duplex-sponge with **rate 64-bit** and **capacity 256-bit**
- ▶ Achieves **security** of **128-bit**

Example2: PHOTON-Beetle (Duplex Sponge Mode with Feedback)



- ▶ Duplex-sponge with a **feedback function ρ** , **rate 128-bit**, **capacity 128-bit**
- ▶ ρ plays the key role to achieve **121-bit security** keeping the capacity to 128-bit

Example3: Oribatida (Sponge with Ciphertext Masking)



- ▶ A sponge with **rate 128-bit** and **capacity 128-bit** with **64-bit ciphertext masking**
- ▶ The masking **boosts** the **security** and ensures resilience in **RUP** settings

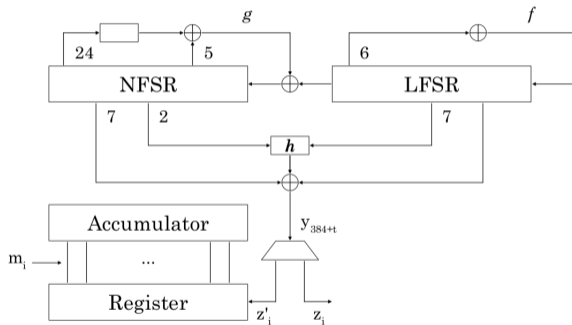
Stream Cipher Modes

- ▶ Use the principle of generating **two key streams** from a short key, and use keystream one for encryption and the other for authentication.
- ▶ The encryption function simply **adds the encryption keystream to the message stream** to generate a ciphertext stream.
- ▶ The authentication module takes the **authentication key stream, message stream or ciphertext stream** to generate the **tag** using universal hash, shift register, permutation, or block cipher modes

Target Applications

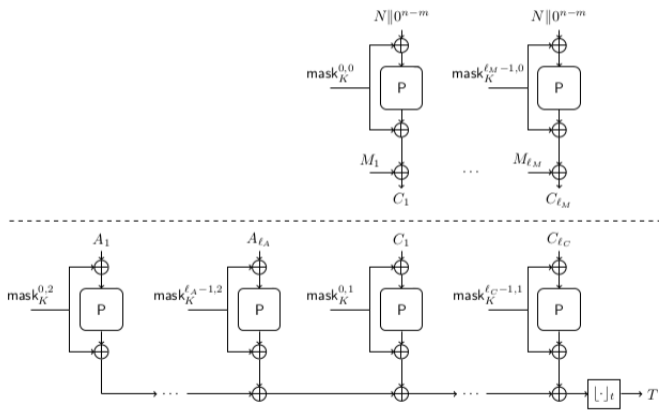
- ▶ Designed specifically to **speed up** the data process as well as to have **low circuit complexity**. More precisely, these designs target to achieve high area efficiency maintaining a high speed.
- ▶ Best choice for applications where **plaintext** comes **with unknown length** like a secure wireless connection.
- ▶ Excellent choice to process **long messages**.

Example1: Grain AEAD



- ▶ Adopts the design of Grain-128 and Grain v1 and extends it for authentication

Example2: Elephant



- Use **public-permutation** to generate the key stream

Classification of NIST Round-2 Candidates

- ▶ **Parallel Mode:** LOTUS-AEAD and LOCUS-AEAD, PAEF (ForkAE), Pyjamask, SKINNY-AEAD
- ▶ **Feedback based Mode:** Comet, GIFT-COFB, HyENA, mixFeed, Romulus-N, SAEAES, SAEF (ForkAE), TinyJAMBU
- ▶ **SIV Mode:** ESTATE, Romulus-M, Sundae-GIFT
- ▶ **Sponge Mode:** ACE, Ascon, DryGASCON, Gimli, ISAP, KNOT, Orange, Oribatida, PHOTON-Beetle, Sparkle, Spix, Spoc, Spook, Subterrain, Wage, Xoodyak
- ▶ **Stream Cipher Mode:** Elephant, Grain-128 AEAD, Saturnin

Thank You..!!! Questions???