Lightweight Cryptography and Classification of AEAD Modes

Nilanjan Datta

Institute for Advancing Intelligence (IAI), TCG CREST

International Crypto-Webniar 2020 **Aug 30, 2020**

tcg crest

Inventing Harmonious Future

Content

- Introduction to Authenticated Encryption
- Motivation of Lightweight Cryptography
- A Discussion on NIST LwC Project
- Classification of lightweight AEAD Modes

I: An Introduction to Authenticated Encryption

The Popular Story: **Encryption**

1. Alice and Bob share a secret key K

The Popular Story: **Encryption**

1. Alice and Bob share a secret key K
2. Alice sends the ciphertext $C = Enc_K(M)$ corresponding to a message M to Bob

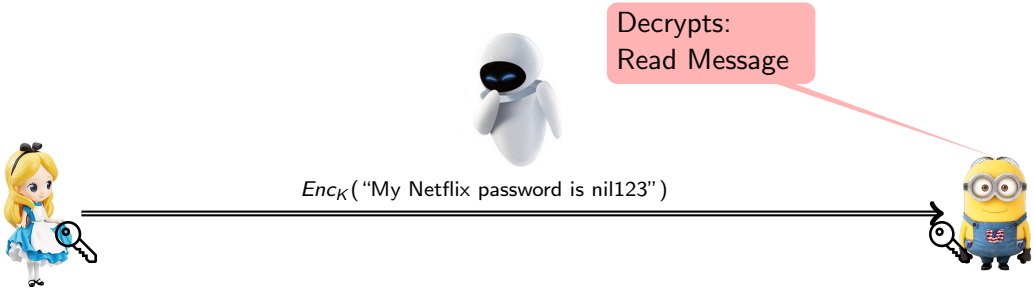
The Popular Story: **Encryption**

1. Alice and Bob share a secret key K
2. Alice sends the ciphertext $C = Enc_K(M)$ corresponding to a message M to Bob
3. Data Privacy: Only Bob can decrypt. No information (other than length) about plaintext is leaked from ciphertext

The Popular Story: **Encryption**

$Enc_K(\text{“My Netflix password is nil123”})$

The Popular Story: **Encryption**



The Popular Story: **Authentication**

1. Alice and Bob share a secret key K

The Popular Story: **Authentication**

1. Alice and Bob share a secret key K
2. Alice sends M along with tag $T = Tag_K(M)$ to Bob

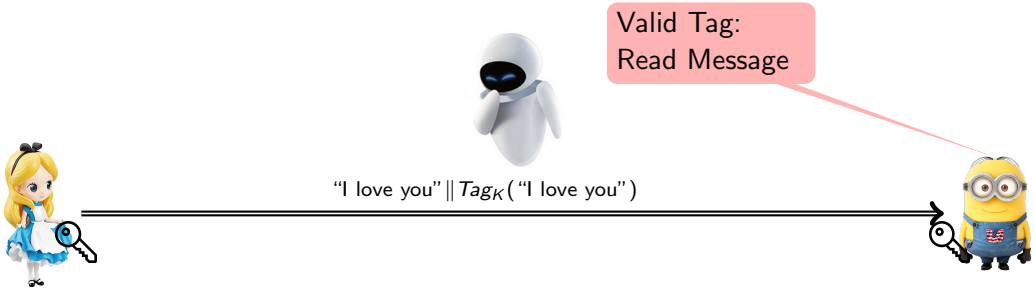
The Popular Story: **Authentication**

1. Alice and Bob share a secret key K
2. Alice sends M along with tag $T = Tag_K(M)$ to Bob
3. Data Integrity: Bob can verify valid tag and can reject all tampered tag

The Popular Story: **Authentication**

"I love you" $\|\, Tag_K(\text{"I love you"})$

The Popular Story: **Authentication**

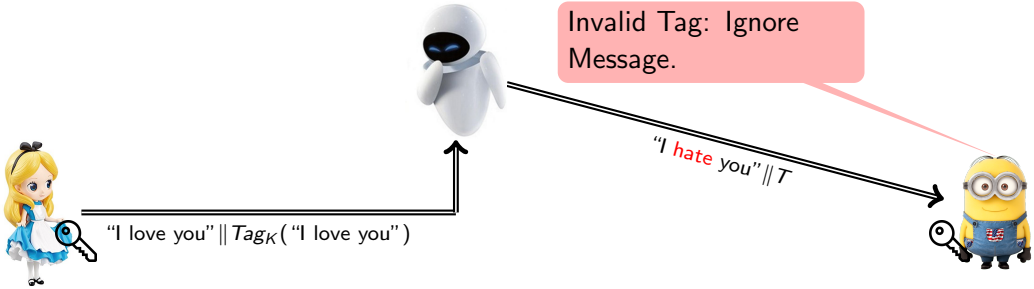


The Popular Story: **Authentication**

"I love you" $\| Tag_K($ "I love you" $)$

"I hate you" $\| T$

The Popular Story: **Authentication**



Authenticated Encryption

Authentication + Encryption

Why Authenticated Encryption?

Chasing the Gangstar

Inspector Alice wishes sub-inspector Bob to move his check point to a different location

Why Authenticated Encryption?

Chasing the Gangstar

Inspector Alice wishes sub-inspector Bob to move his check point to a different location

- Data *Privacy*: The updated check point should remain *confidential*

Why Authenticated Encryption?

Chasing the Gangstar

Inspector Alice wishes sub-inspector Bob to move his check point to a different location

- Data *Privacy*: The updated check point should remain *confidential*
- Data *Authenticity*: Alice sends the updated check point and it has not been modified in transit

Why Authenticated Encryption?

Covid Report

Doctor Alice wishes to send the covid report M of Bob to the medical database

Why Authenticated Encryption?

Covid Report

Doctor Alice wishes to send the covid report M of Bob to the medical database

- Data *Privacy*: Bob's medical records should remain *confidential*

Why Authenticated Encryption?

Covid Report

Doctor Alice wishes to send the covid report M of Bob to the medical database

- Data *Privacy*: Bob's medical records should remain *confidential*
- Data *Authenticity*: Alice prepares the report and the report was not modified in transit

Why Authenticated Encryption?

The Annual Evaluation

Prof. Alice wishes to inform the semestral marks to her students

- Data *Privacy*: Bob's marks should remain *confidential*
- Data *Authenticity*: Alice sends the marks and the marks have not been modified in transit

Authenticated Encryption with Associated Data (AEAD)

AEAD Algorithm

$$\mathcal{AE}(K, AD, M, N) \rightarrow (C, T)$$

Authenticated Encryption with Associated Data (AEAD)

AEAD Algorithm

$$\mathcal{AE}(K, AD, M, N) \rightarrow (C, T)$$

What is Associated Data?

- *Header* of the Message. Example: *IP Address*
- Requires data authenticity, not privacy

Authenticated Encryption with Associated Data (AEAD)

AEAD Algorithm

$$\mathcal{AE}(K, AD, M, N) \rightarrow (C, T)$$

What is Associated Data?

- *Header* of the Message. Example: *IP Address*
- Requires data authenticity, not privacy

What is Nonce?

- An arbitrary number used only *once*. Example: *Counter*
- Used to generate randomness

Authenticated Encryption with Associated Data (AEAD)

Verified Decryption Algorithm

$$\mathcal{VD}(K, AD, C, T, N) \rightarrow M/\perp.$$

Authenticated Encryption with Associated Data (AEAD)

Verified Decryption Algorithm

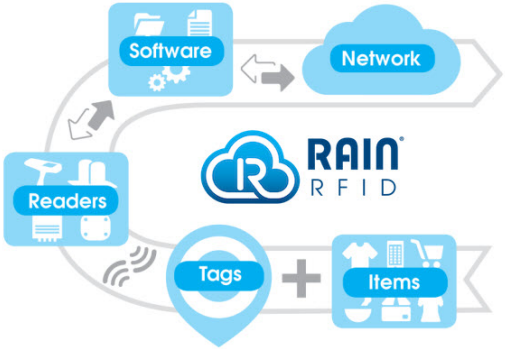
$$\mathcal{VD}(K, AD, C, T, N) \rightarrow M/\perp.$$

A Note on Verified Decryption

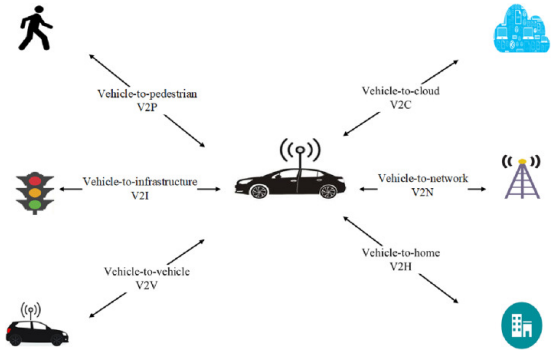
- Plaintext is only released after verification is successful
- Otherwise, the algorithm aborts
- However, the ordering of verification and decryption may vary

II: An Introduction to Light-weight Cryptography

Lightweight Cryptography: Use in Rain RFID Tags

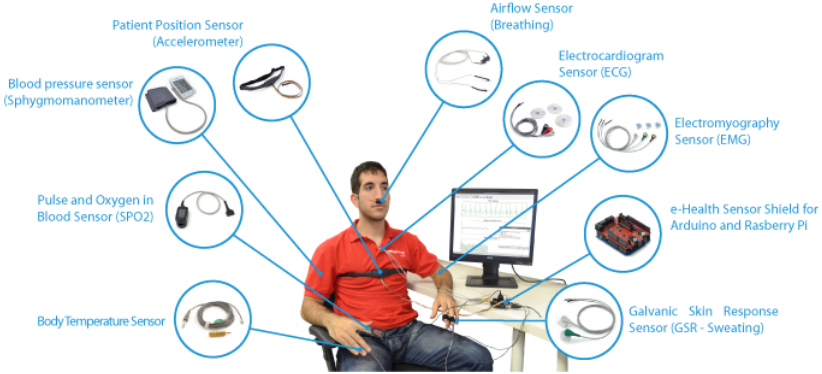


Lightweight Cryptography: Use in Vehicles



Lightweight Cryptography: Use in Smart Home

Lightweight Cryptography: Use in Medical Sensors



What is Lightweight Cryptography?

Lightweight Cryptography

Subfield of Cryptography that aims to provide crypto solutions tailored to constrained environments

What is Lightweight Cryptography?

Lightweight Cryptography

Subfield of Cryptography that aims to provide crypto solutions tailored to constrained environments

$$\text{Lightweight} = \text{Light} + \text{Weight}$$

What is Lightweight Cryptography?

Lightweight Cryptography

Subfield of Cryptography that aims to provide crypto solutions tailored to constrained environments

$$\text{Lightweight} = \text{Light} + \text{Weight}$$

Weight of an Algorithm

A property of its implementation depending on different metrics of the target platform

Weight of An Algorithm

Weight of An Algorithm

Weight of An Algorithm

Weight of An Algorithm

Weight of An Algorithm

General Purpose Crypto vs Lightweight Crypto

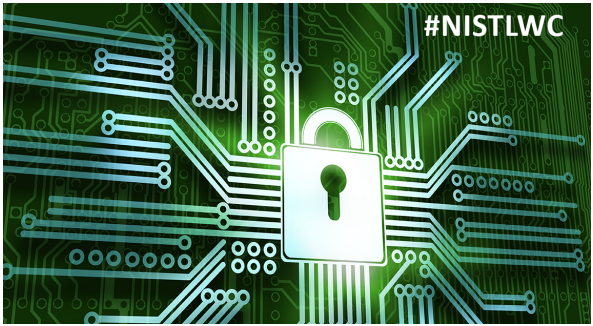
General Purpose Crypto

- Used in several applications
- A proper trade-off of various metric: area, speed, throughput, energy etc

Lightweight Crypto

- Used for dedicated resource constraint environment
- Lack of Crypto standards suitable for such devices.

III: A Brief Overview on NIST LwC Project



NIST LwC Project: Timeline

- Aug 2018: Call for Algorithms
- Mar 2019: First Round Submission
- Aug 2019: Declaration of Second Round Candidates
- Sep 2019: Second Round Submission
- Dec 2020: Declaration of Third Round Candidates
- End of 2021: Final Portfolio

NIST LwC Project: Requirements

- Perform significantly better in constrained environments (HW and SW platforms)
- Efficient for short messages
- Implementations should have cheap countermeasures against side channel attacks, and fault attacks
- Scope: AEAD with optional hashing functionality

NIST LwC First Round Candidates

- Total 57 submissions in Round 1 (12 from India)

*Figure Courtesy: Meltem Sonmez Turan

NIST LwC Submission Statistics

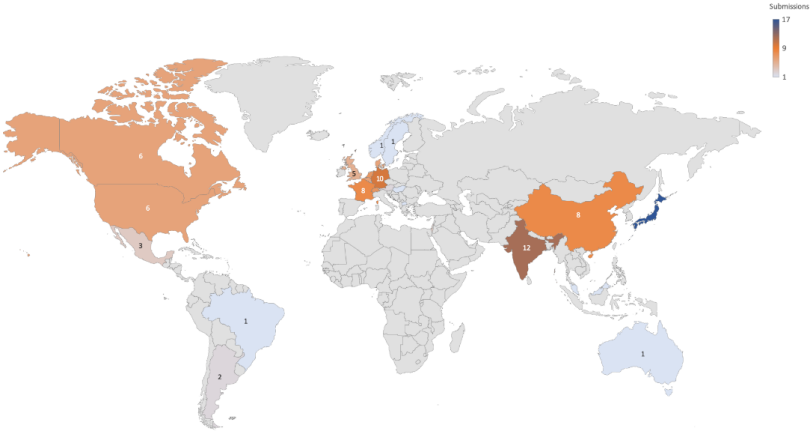


*Figure Courtesy: Meltem Sonmez Turan

NIST LwC Second Round Candidates

- Total 32 candidates moved to Round 2 (10 from India)

ACE	ASCON	COMET	DryGASCON	Elephant	ESTATE
ForkAE	GIFT-COFB	Gimli	Grain-128 AEAD	HyENA	ISAP
KNOT	LOTUS-AEAD & LOCUS-AEAD	mixFeed	ORANGE	Oribatida	PHOTON-Beetle
Pyjamask	Romulus	SAEAES	Saturnin	SKINNY-AEAD	SPARKLE
Spix	SpoC	Spook	Subterrain 2.0	SUNDAE-GIFT	TinyJAMBU
WAGE	Xoodyak				

NIST LwC Second Round Candidates

- Total 32 candidates moved to Round 2 (10 from India)

ACE	ASCON	COMET	DryGASCON	Elephant	ESTATE
ForkAE	GIFT-COFB	Gimli	Grain-128 AEAD	HyENA	ISAP
KNOT	LOTUS-AEAD & LOCUS-AEAD	mixFeed	ORANGE	Oribatida	PHOTON-Beetle
Pyjamask	Romulus	SAEAES	Saturnin	SKINNY-AEAD	SPARKLE
Spix	SpoC	Spook	Subterrain 2.0	SUNDAE-GIFT	TinyJAMBU
WAGE	Xoodyak				

All the 10 submissions are co-designed by **Prof. Mridul Nandi**

IV: Lightweight AEAD: Features and Design

Some Relevant Features of AEAD Modes

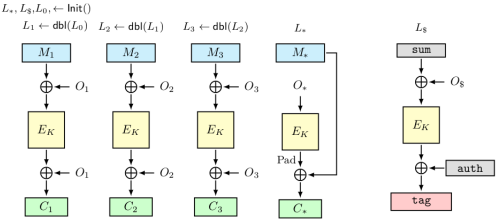
Single-Pass

- Make only one pass through the data, simultaneously doing what is needed to ensure both privacy and authenticity
- Computational cost for single-pass schemes are about half of a two-pass scheme, and hence efficient

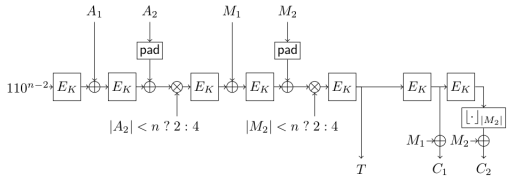
Some Relevant Features of AEAD Modes

Single-Pass

- Make only one pass through the data, simultaneously doing what is needed to ensure both privacy and authenticity
- Computational cost for single-pass schemes are about half of a two-pass scheme, and hence efficient



Single Pass



Two Pass

Some Relevant Features of AEAD Modes

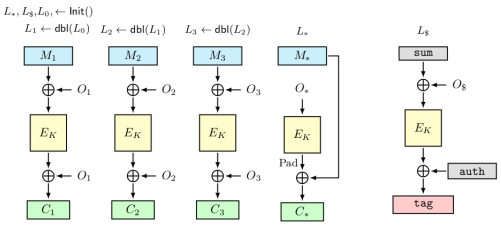
On-line

- Encryption produces cipher-text blocks on the fly, and before subsequent plain-text blocks are known
- Ensures that no additional memory is needed to store intermediate results for later use
- Useful in real-time streaming protocols as it reduces the end-to-end latency

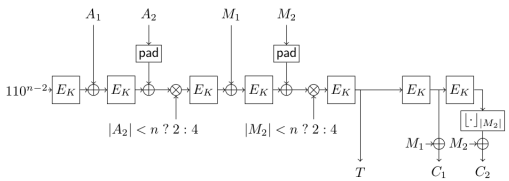
Some Relevant Features of AEAD Modes

On-line

- Encryption produces cipher-text blocks on the fly, and before subsequent plain-text blocks are known
- Ensures that no additional memory is needed to store intermediate results for later use
- Useful in real-time streaming protocols as it reduces the end-to-end latency



On-line



Not On-line

Some Relevant Features of AEAD Modes

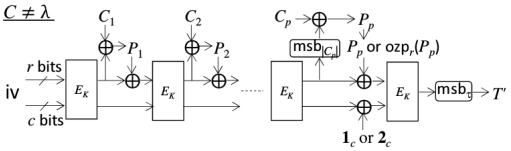
Parallel

- All the ciphertext blocks can be computed in parallel
- Allows both H/W and S/W acceleration proportional to the available computational unit
- Can have a fully pipelined implementation, reduces latency and provides high speed

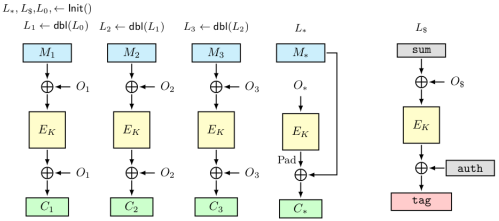
Some Relevant Features of AEAD Modes

Parallel

- All the ciphertext blocks can be computed in parallel
- Allows both H/W and S/W acceleration proportional to the available computational unit
- Can have a fully pipelined implementation, reduces latency and provides high speed



Sequential



Parallel

Some Relevant Features of AEAD Modes

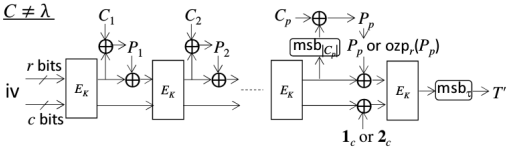
Inverse-Free

- Both the encryption and verified decryption algorithm does not invoke the inverse of the primitive.
- Saves significant area in combined encryption-decryption AEAD implementation

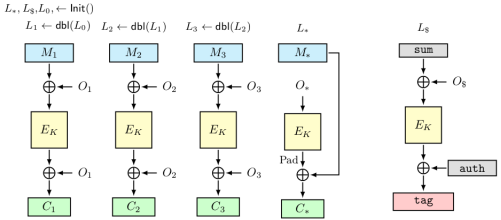
Some Relevant Features of AEAD Modes

Inverse-Free

- Both the encryption and verified decryption algorithm does not invoke the inverse of the primitive.
- Saves significant area in combined encryption-decryption AEAD implementation



Inverse-free



Not Inverse-free

Some Relevant Features of AEAD Modes

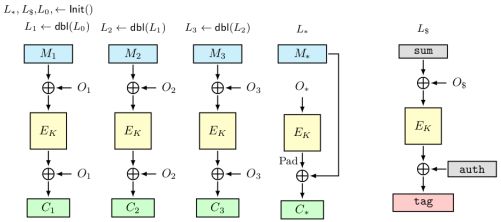
State-size

- A theoretic estimate of the register size
- Directly corresponds to the size of memory
- State size should be as low as possible for area efficient designs

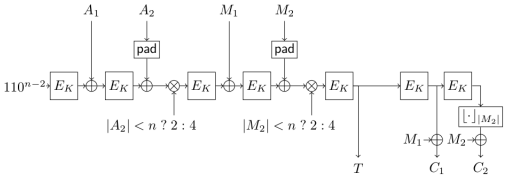
Some Relevant Features of AEAD Modes

State-size

- A theoretic estimate of the register size
- Directly corresponds to the size of memory
- State size should be as low as possible for area efficient designs



State size: $3n + k$ bits



State size: $n + k$ bits

Some Relevant Features of AEAD Modes

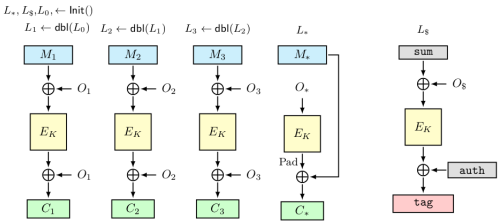
High Rate

- Rate is defined as number of message blocks processed per primitive invocation
- Rate of an AEAD mode can be at most 1
- Constructions with higher rate reduces latency, and particularly beneficial to obtain higher speed

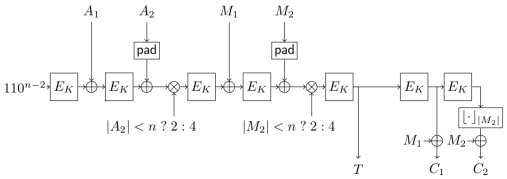
Some Relevant Features of AEAD Modes

High Rate

- Rate is defined as number of message blocks processed per primitive invocation
- Rate of an AEAD mode can be at most 1
- Constructions with higher rate reduces latency, and particularly beneficial to obtain higher speed



Rate: 1



Rate: 0.5

Some Relevant Features of AEAD Modes

Optimal (Primitive Invocation)

- Uses the minimum possible number of non-linear invocations
- The bound is given in [Chakraborti et al., JMC 18]:
  - Nonce based AEAD: $(a + m + 1)$
  - Deterministic: $(a + 2m)$
- Makes a construction efficient for short messages and reduces the latency

Some Relevant Features of AEAD Modes

Optimal (Primitive Invocation)

- Uses the minimum possible number of non-linear invocations
- The bound is given in [Chakraborti et al., JMC 18]:
  - Nonce based AEAD: $(a + m + 1)$
  - Deterministic: $(a + 2m)$
- Makes a construction efficient for short messages and reduces the latency

Efficient Static AD Processing

- In many scenarios, AD remain static over the course of a communications session
- Efficient static AD processing use a pre-processed computed value instead of whole computation

Some Relevant Features of AEAD Modes

Nonce Misuse Resistance

- Provides security even if nonce is repeated, or even without nonce
- Well suited for lightweight applications where storing counter or generating random number may be difficult to implement

Some Relevant Features of AEAD Modes

Nonce Misuse Resistance

- Provides security even if nonce is repeated, or even without nonce
- Well suited for lightweight applications where storing counter or generating random number may be difficult to implement

Integrity under RUP

- Verified Decryption: Plaintext is released after verification
- Small buffer size may force decryption algo to release plaintext before verification
- This gives an adversary additional power, which may be exploited for forging

Mode Classification

- Parallel Mode
- Feedback based Mode
- SIV Mode
- Sponge Mode
- Stream Cipher Mode

Parallel Modes

- Inputs of the block ciphers depend on the message and not on the previous block cipher outputs or cipher texts, hence parallelization in the computation between distinct block cipher calls
- Typically Used in low-latency scenarios as well as for obtaining good performance from both high-speed hardware and commodity processors
- The parallel design allows to efficiently process subsequent message blocks exploiting the CPU pipeline and multi-threading techniques

Parallel Modes: Design Principle

- The design principles follow the ECB structure
- To ensure (i) privacy within a message and (ii) privacy within two messages, some additional masking state depending on the value of the nonce and the block number

Parallel Modes: Design Principle

Typical Choices:

- Xor-Encrypt-Xor paradigm
- tweakable block ciphers with tweak defined as a pair (nonce, block number)

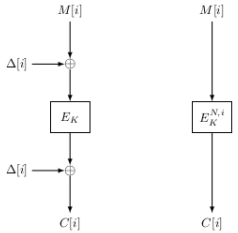
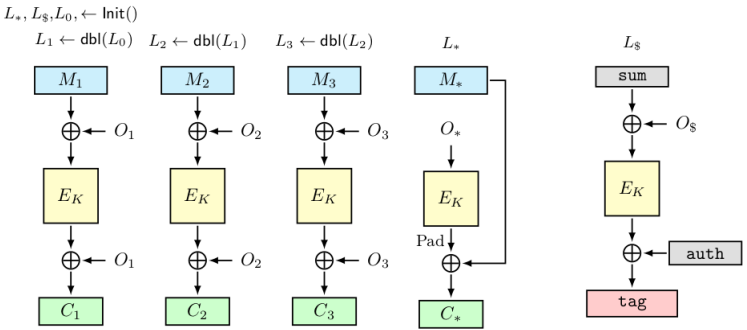


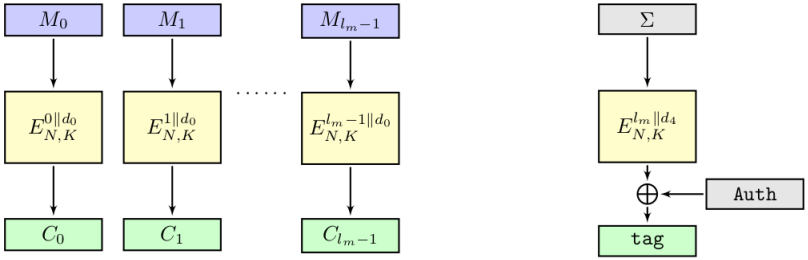
Figure: Parallel Mode of Encryption: (a) OCB, (b) ΘCB

Example1: Pyjamask (OCB Style Encryption)



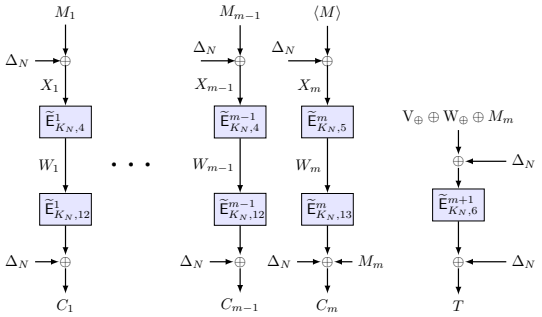
- Block Cipher based, Parallel, Online, Rate-1
- Birthday bound secure, No RUP Security

Example2: Skinny AEAD (ΘCB Style Encryption)



- Tweakable Block Cipher based, Parallel, Online, Rate-1
- Birthday bound secure, No RUP Security

Example3: LOCUS-AEAD (OCB Style with Intermediate checksum)

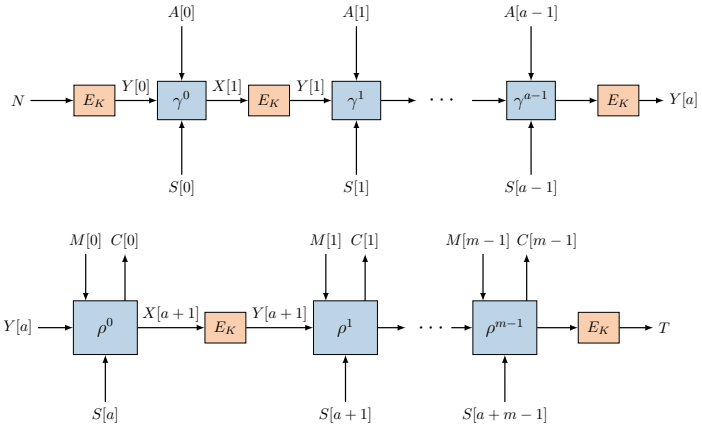


- Short-tweak TBC based, Parallel, Online
- Nonce-derived key for full security, Intermediate checksum to achieve RUP security

Feedback based Modes

- One of the most popular method of constructing area-efficient block cipher based AE
- 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
- Typically inverse-free and area-efficient for combined enc-dec implementations

Basic Structure of the Mode



Types of Feedback

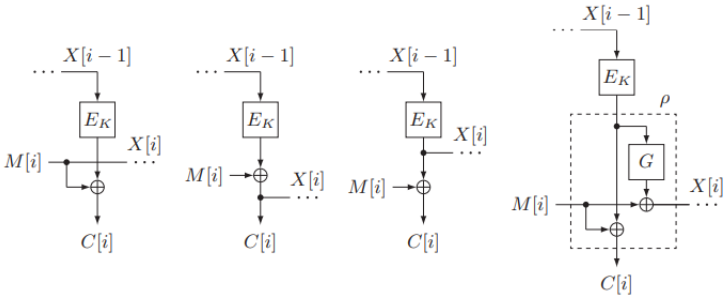


Figure: Hybrid Feedback functions: (a) PFB, (b) CFB, (c) OFB, (d) CoFB

Types of Hybrid Feedback

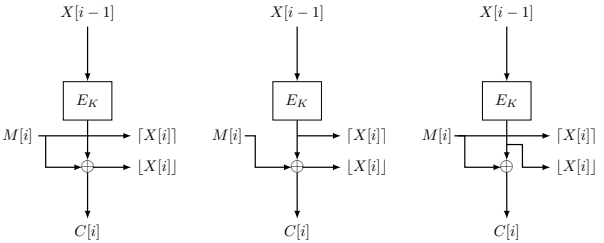


Figure: (a) PFB+CFB, (b) OFB+CFB, (c) OFB+PFB

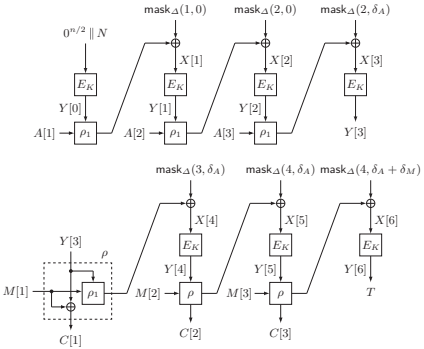
Investigating the Security of Rate-1 Feedback based AE

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]

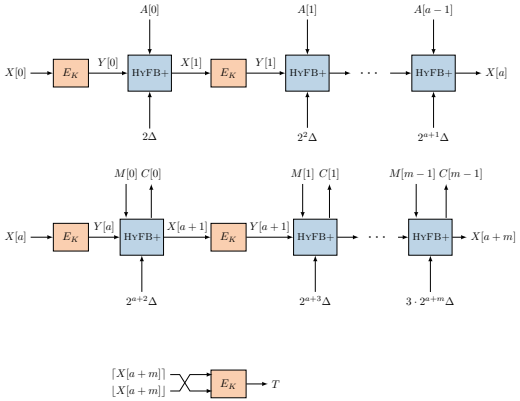
For any rate-1 feedback-based AE with additional state of τ-bits, there is an adversary that breaks the construction with query complexity 2^{τ}

Example1: COFB (A Mode with Combined Feedback)



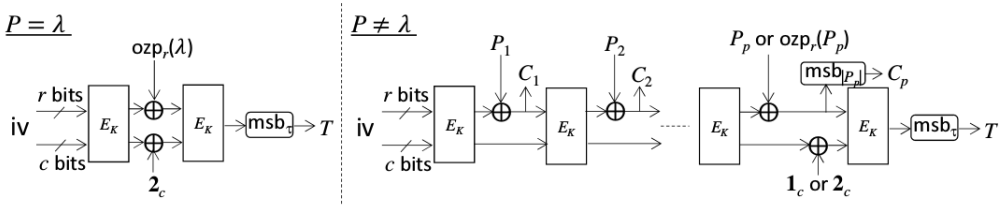
- Inverse-free, Rate-1, Efficient AD Processing
- State size: $1.5n$-bit (optimal for rate-1), XOR: $2n$-bit

Example2: HyENA (A Mode with Hybrid Feedback)



- Inverse-free, Rate-1, Efficient AD Processing
- State size: $1.5n$-bit (optimal for rate-1), XOR: n-bit

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



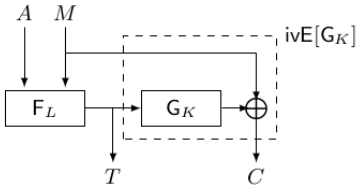
- Inverse-free, Rate-1/2, Efficient AD Processing
- State size: n-bit (optimal irrespective of rate)

SIV based Modes

- Provide maximal robustness to a lack of proper randomness or secure state
- Follows MAC-then-Encrypt structure, and hence two pass mode
- Typically obtain single-state implementation
- Excellent choice for short message processing

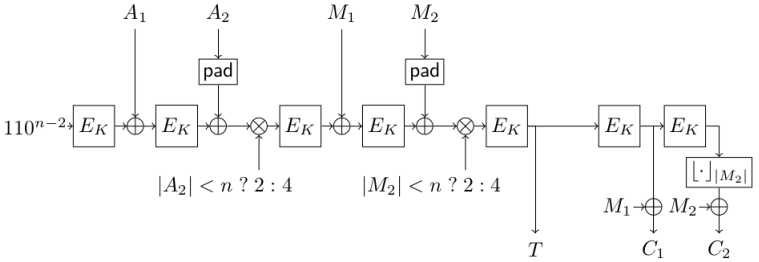
Design Principle

- Follow the MAC-then-ENCRYPT structure



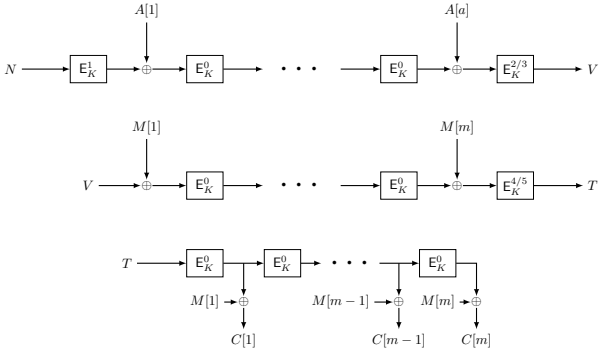
- A typical choice is CBC-type MAC (single-state implementation) follows by OTP or OFB with the tag as IV

Example1: SUNDAE



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

Example2: ESTATE

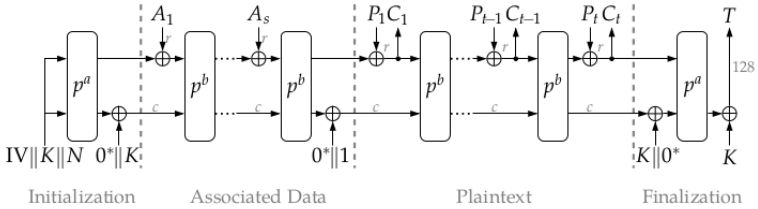


- Single-state, Inverse-free, efficient for short messages
- Use tBC to ensure RUP security, optimal block-cipher invocations

Sponge Modes

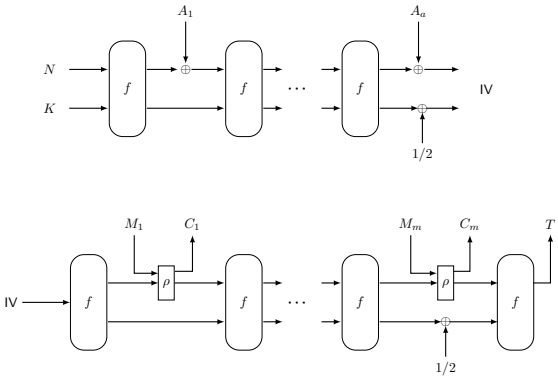
- Use public permutation instead of keyed permutation
- Employs duplex mode of operation - absorbs the data and then squeeze the ciphertext
- Has the advantage of key agility: no key-scheduling

Example1: ASCON (Simple Duplex type Sponge Mode)



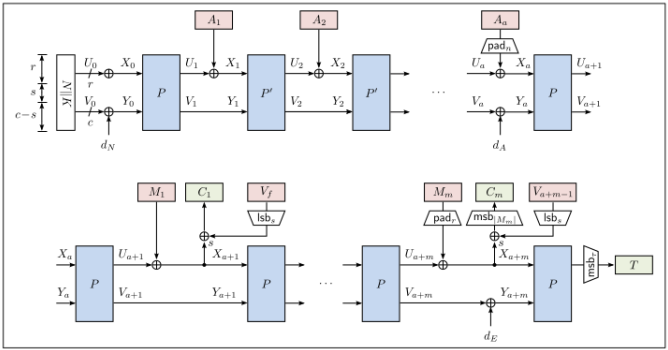
- Use simple duplex-sponge mode of operation with sponge rate 64-bit and sponge capacity 256-bit
- Achieves security of 128-bit

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



- Uses duplex-sponge mode with a feedback function ρ with sponge rate 128-bit and sponge capacity 128-bit
- ρ plays the key role to achieve 121-bit security keeping the sponge capacity to 128-bit

Example3: Oribatida (Sponge with Ciphertext Masking)

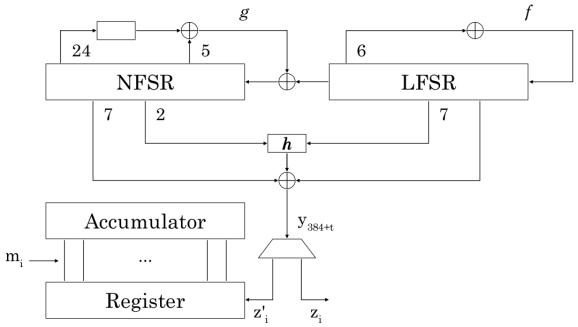


- A sponge with sponge rate 128-bit and sponge capacity 128-bit with 64-bit ciphertext masking
- The masking boosts the security and ensures resilience in RUP settings

Stream Cipher Modes

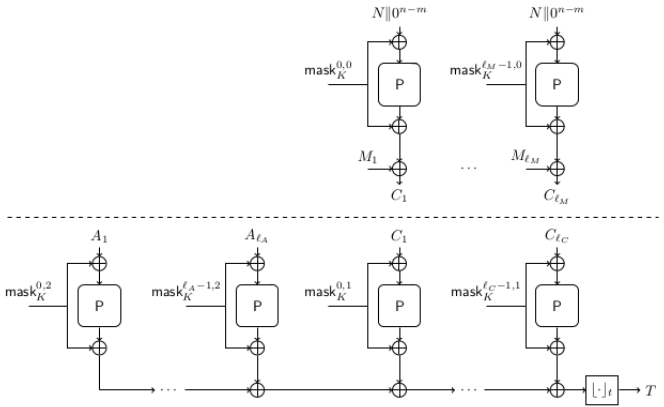
- Use a stream-cipher encryption as the basic mode
- Used to design fast and energy-efficient AEAD
- Typically, no additional states are reserved, and hence area-efficient
- Excellent choice for long messages

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

Summary

- Authenticated Encryption - Motivation, Basic Idea

Summary

- Authenticated Encryption - Motivation, Basic Idea
- Lightweight Cryptography - Motivation

Summary

- Authenticated Encryption - Motivation, Basic Idea
- Lightweight Cryptography - Motivation
- NIST LwC Project - Timeline and Progress

Summary

- Authenticated Encryption - Motivation, Basic Idea
- Lightweight Cryptography - Motivation
- NIST LwC Project - Timeline and Progress
- Classification of AEAD Modes based on Design

Thank You..!!! Questions???

Light-Weight Cipher Design Challenge 2020

Link: https://www.dsci.in/ncoe-light-weight-cipher-design-challenge-2020/