Nonce Misuse-Resistant Authenticated Encryption for Automotive Ethernet

IEEE STANDARDS ASSOCIATION IEEE

2017 IEEE-SA Ethernet & IP @ Automotive Technology Day

Patrick Kresmer, Alexander Zeh
Presenter: Harald Zweck

Infineon Technologies

Infineon

Agenda

- 1 Motivation
- 2 Introduction to AEAD
- 3 Nonce misuse on AES-GCM
- 4 Nonce-misuse-resistant AES-GCM-SIV
- 5 Summary
- 6 Questions

Motivation

What do we need it in the automotive context?

We have	We need
▪ CAN, Ethernet, FlexRay, LIN,.. ▪ Gateways, Bridges V2X, .. ▪ Tuning Protection, Immobilizer, SOTA,... ▪ **Bad People**	❑ **Authenticity** ❑ **Integrity** ❑ **Confidentiality**

=> We need **Authenticated Encryption** (= AE)

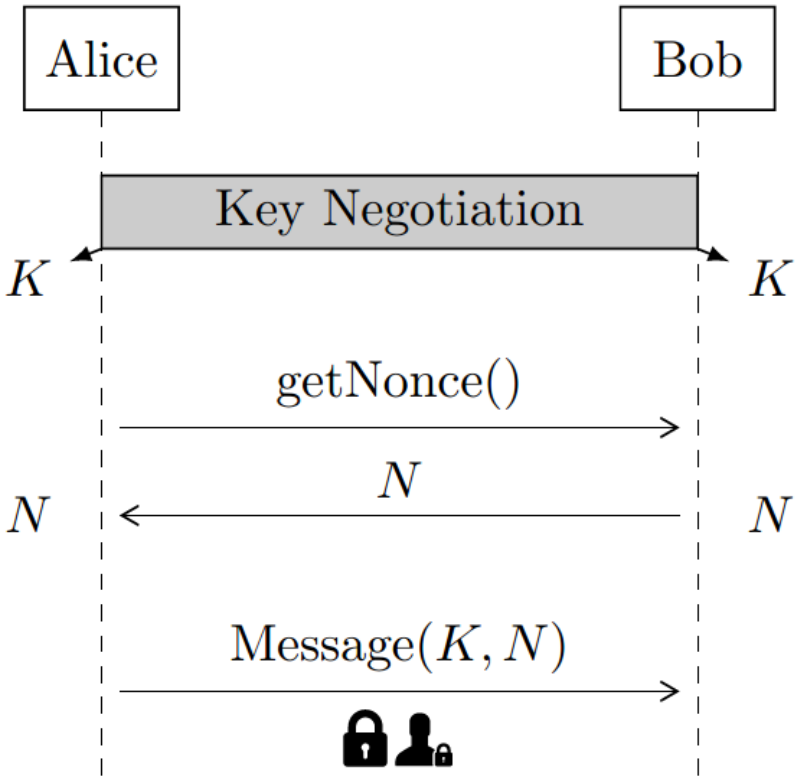
Motivation

Available Cryptographic Tools

	Authenticity	**Integrity**	**Confidentiality**
Block Cipher	✗	✗	✓
Hash	✗	✓	✗
MAC	✓	✓	✗
AE(AD)	✓	✓	✓

$$\underbrace{\text{Authenticated Encryption}}_{\text{Auth., Int., Conf.}}\ \underbrace{\text{(with Associated/Additional Data)}}_{\text{Auth., Int.}}$$

Introduction to AEAD

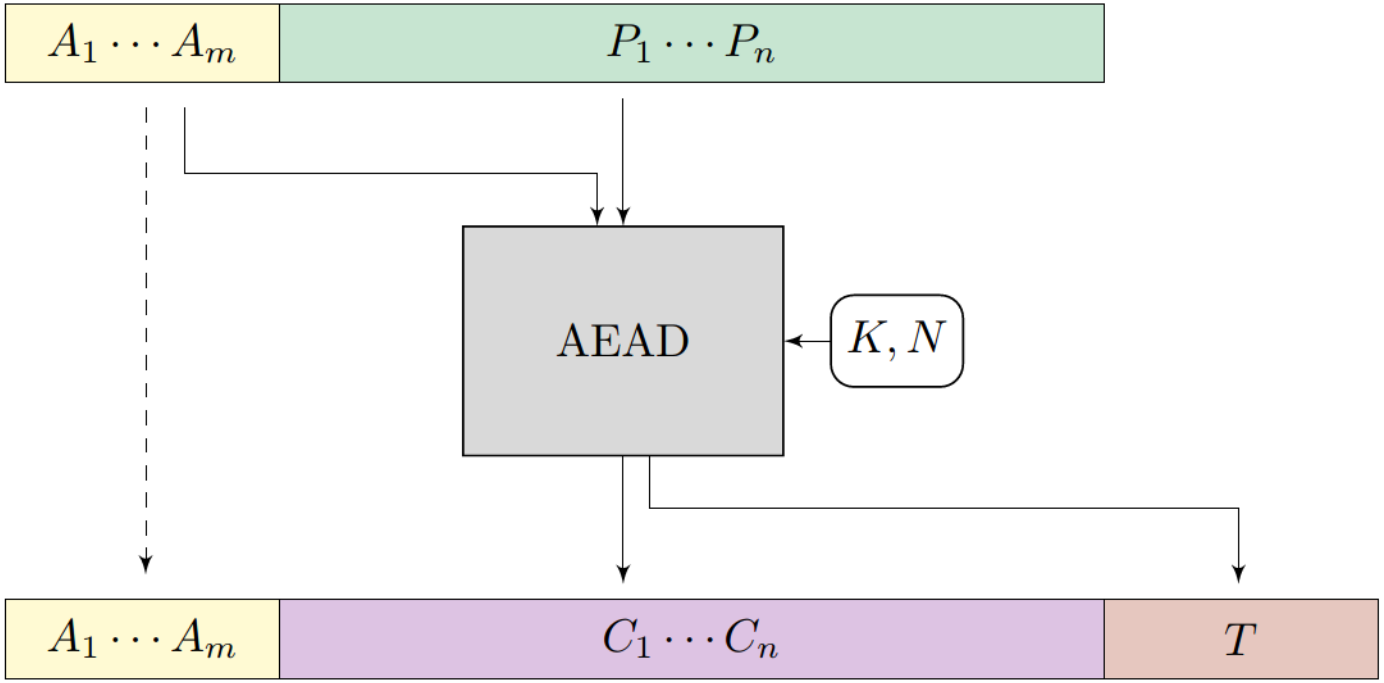


› **Prerequisites for AEAD***

- Secret key *K* was shared
- Unique nonce *N* was shared

AEAD: Authenticated Encryption with Associated Data

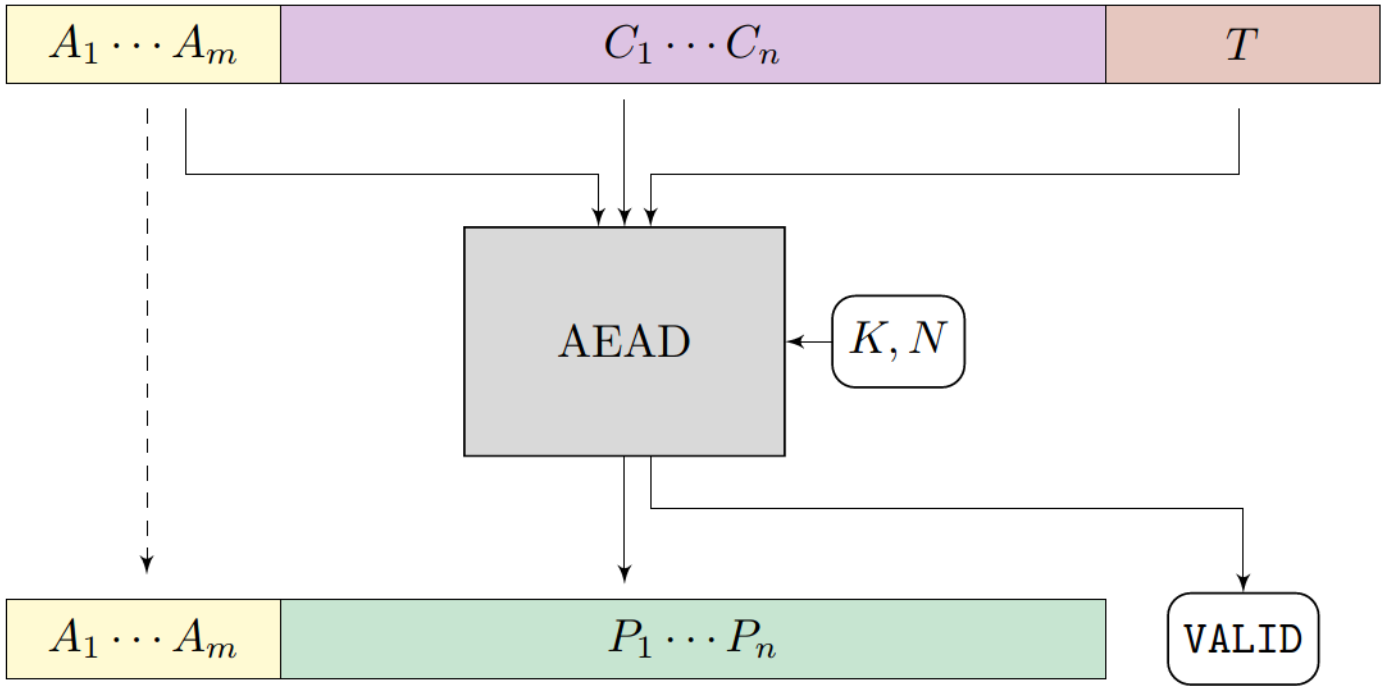
Encryption Flow (Alice)



$A_1 \ldots A_m$: Authenticated data fields 1 .. m
$P_1 \ldots P_n$: Plain text fields 1 .. n
$C_1 \ldots C_n$: Cipher text fields 1 .. n
T: Tag field

K: Key
N: Nonce

Decryption Flow (Bob)

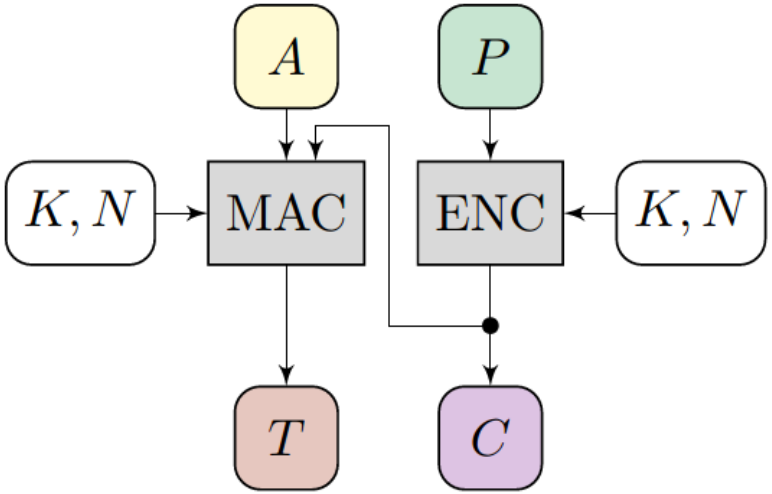


Summary

* Authenticated data field ($A_1..A_m$)
* Plain text encrypted (Confidentiality)
* Both protected by a Tag (Integrity)

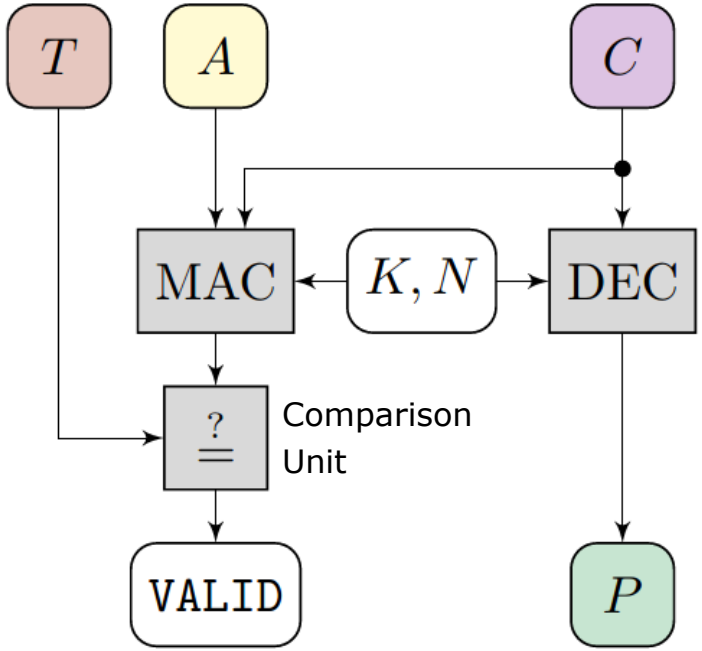
Introduction to AEAD

Authenticated Encryption (Alice)



A: Authenticated data fields
P: Plain text fields
C: Cipher text fields
T: Tag field
K: Key
N: Nonce

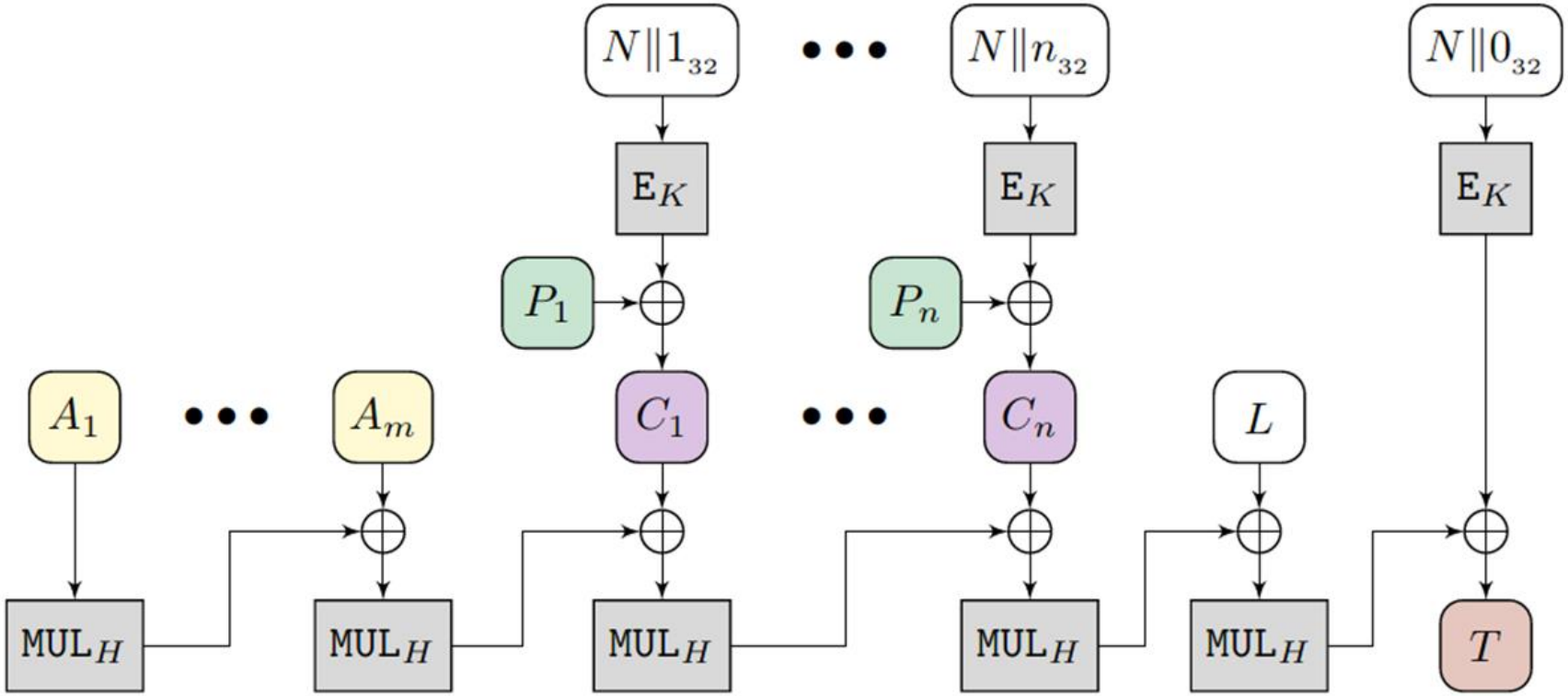
Authenticated Decryption (Bob)



MAC: Message Authentication Code
ENC: Encoder unit
DEC: Decoder unit

Introduction to AES-GCM Encryption* [3]

*AES-GCM: Advanced Encryption Standard – Galois Counter Mode

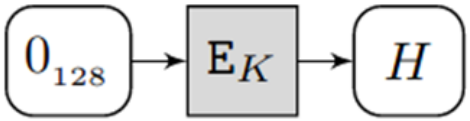


E_K: Encryption using key K
N: Nonce
L: Length of message
⊕ Bit XOR
n_{xy}: n-th field of message
MUL_H see next page

[1] Viega-McGrew2005

Introduction to AES-GCM Encryption

- MUL_H
  - Produced by encrypting the value “0”



- Function of MUL_H

MUL_H: multiplication by H in $\mathbb{F}_{2^{128}} \simeq \mathbb{F}_2[x]/(x^{128} + x^7 + x^2 + x + 1)$

Nonce Misuse on AES-GCM

Threat analysis:
Impact if nonce *N* is **not unique** for given key *K* ? [4]

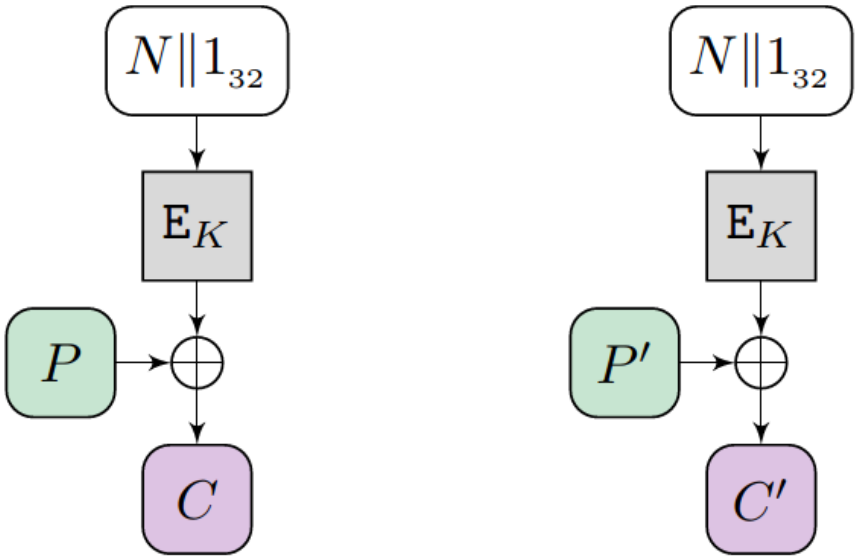
Possible root causes

- Software bug (always same *N*)
- Counter overflow (after 2^8 messages for uint8 *N*)
- Random Nonce (50% chance after ~1.2 * $\sqrt{2^{\omega}}$ messages)
- Attacker's intention (Man-in-the-Middle scenario)

[2] Böck et al. 2016

Attack 1: Plaintext Leak

Attacker's analysis of two distinct messages under same K, N:

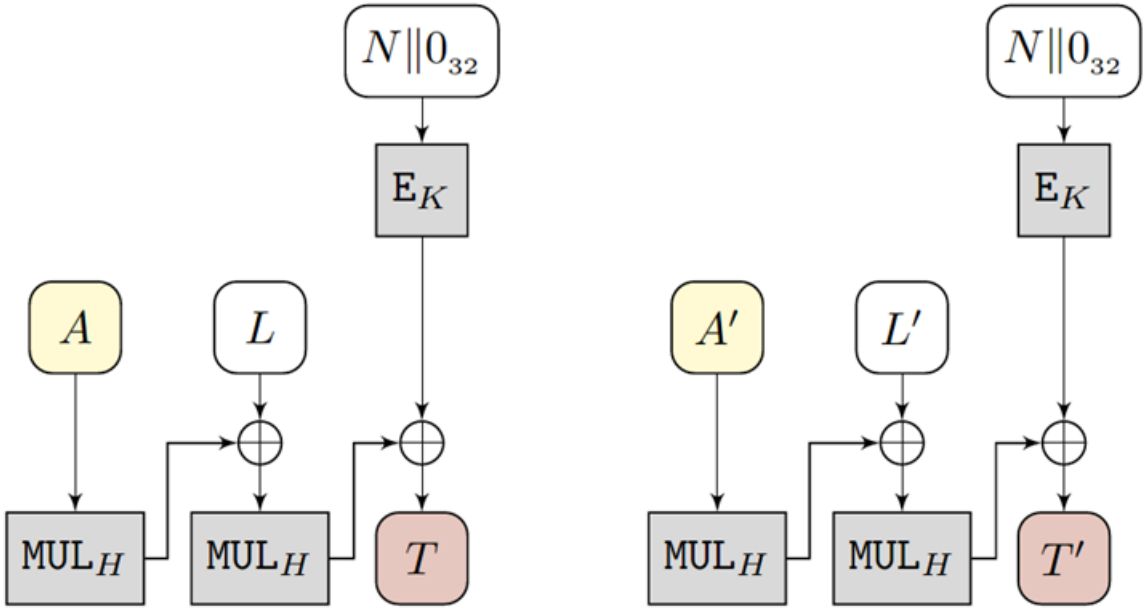


$$C \oplus C' = P \oplus \mathrm{E}_K(N) \oplus P' \oplus \mathrm{E}_K(N) = P \oplus P'.$$

⇒ Attacker learns the difference between two different messages

⇒ Confidentiality is broken

Attack 2: Key Recovery [5]



$$T \oplus T' = f(H) \oplus \mathrm{E}_K(N) \oplus f'(H) \oplus \mathrm{E}_K(N) = p(H).$$

$$p(H) = (A + A')H^2 + (L + L')H.$$

⇒ For simplification reasons plain text fields are not used

⇒ N||0_{32} -> 96 bit Nonce and 32 bit counter (which provides the value 0)

[3] A. Joux 2006

Attack 2: Key Recovery

- Flow
  - If the nonce is the same, a polynomial *p(H)* can be built.
  - Solving the polynomial gives the key *H*.
  - If the key *H* is known, the Tag *T* can be calculated.
  - Knowing Tag *T* means that “valid” messages can be “forged”.

  => Authenticity and integrity is broken.

Nonce Misuse

- Summary
- Repeating Nonce(s) is fatal for AES-GCM:

  Attack 1: confidentiality broken

  Attack 2: authenticity, integrity broken (forgery)

- **Crucial point**
  - The Nonce *N* is the only input for PRF E_K (for given *K*)
  - XOR-ing distinct messages cancels out key stream

Synthetic Initialization Vector

Solution

- **Synthetic Initialization Vector** (SIV) construction [6] :

- Use *N* and message as input for PRF* to compute T
  - Different key streams for distinct messages ($T \oplus T' = ?$)
  - Same T for same messages ($T \oplus T' = 0$)

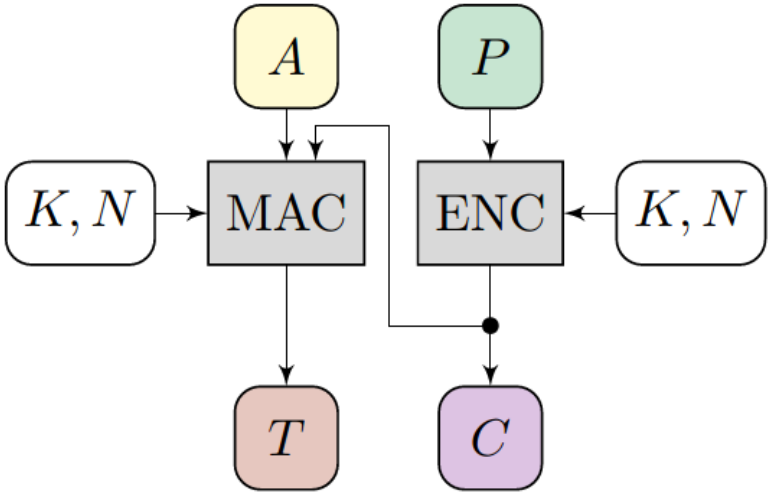
- Use T as (synthetic) IV for PRF to compute *C*
  - Different key streams for distinct messages ($C \oplus C' = ?$)
  - Same C for same messages ($C \oplus C' = 0$)

[4] Rogaway –Shrimpton 2006

**PRF*: Pseudo Random Function

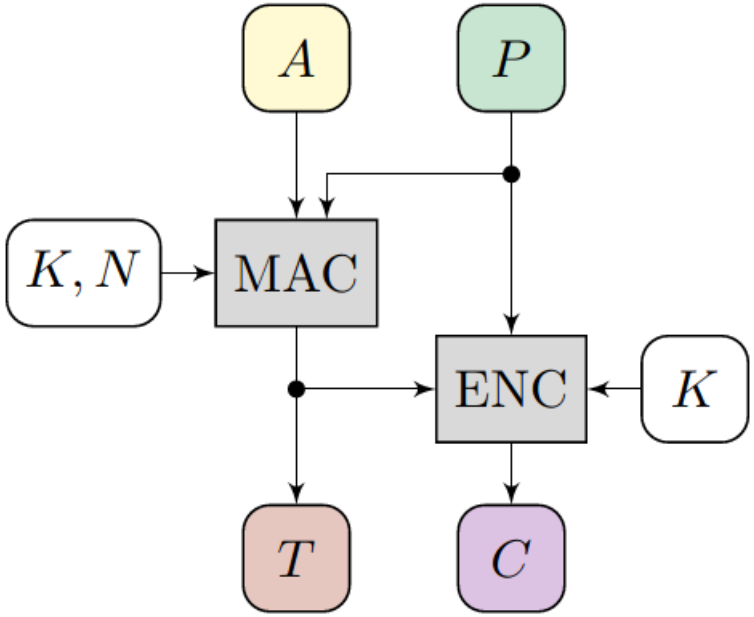
Synthetic Initialization Vector

Authenticated Encryption *Old*



⇒ Output ENC based on *P, K, N*

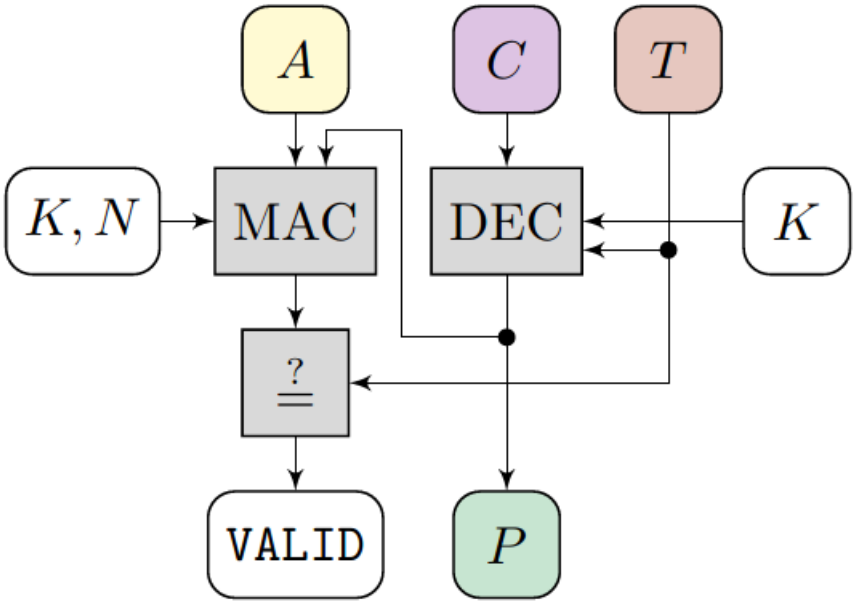
Authenticated Encryption *New*



⇒ Output ENC based on *P, K, T*

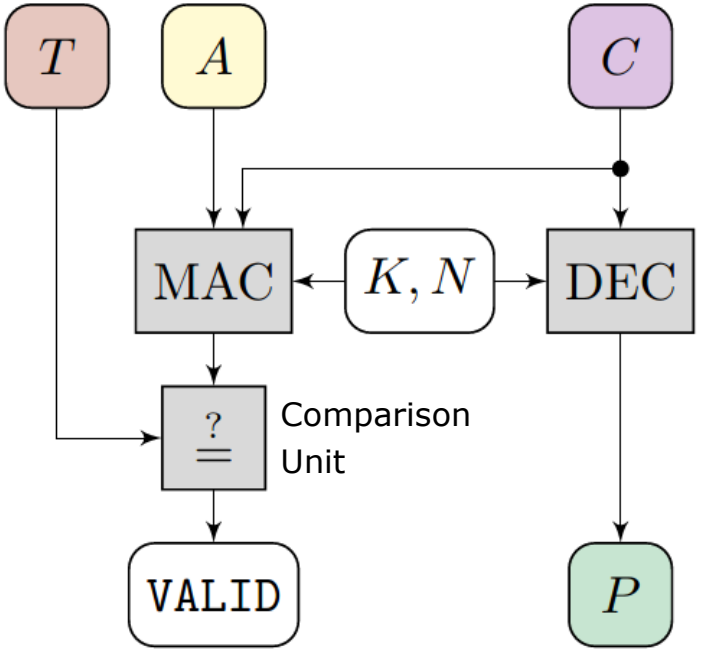
Synthetic Initialization Vector

Authenticated Decryption *New*



⇒ Output DEC based on *C, K, T*

Authenticated Decryption *Old*

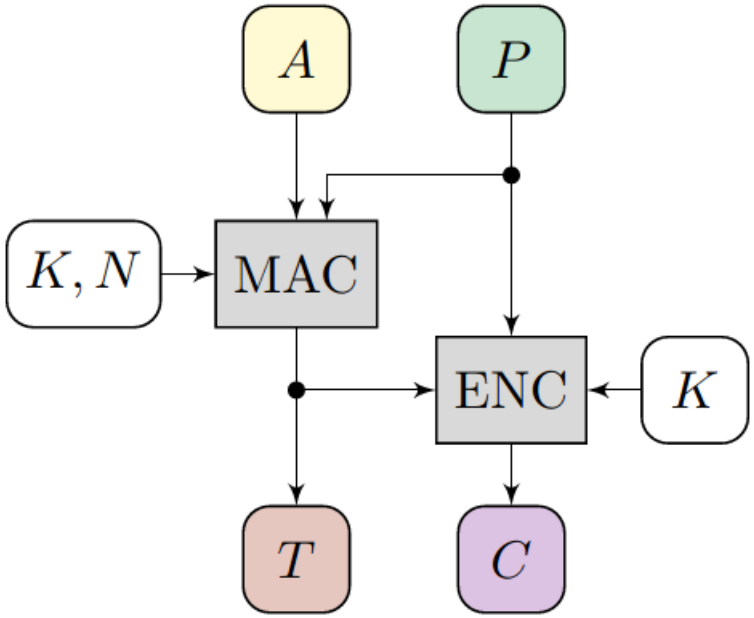


⇒ Output DEC based on *C, K, N*

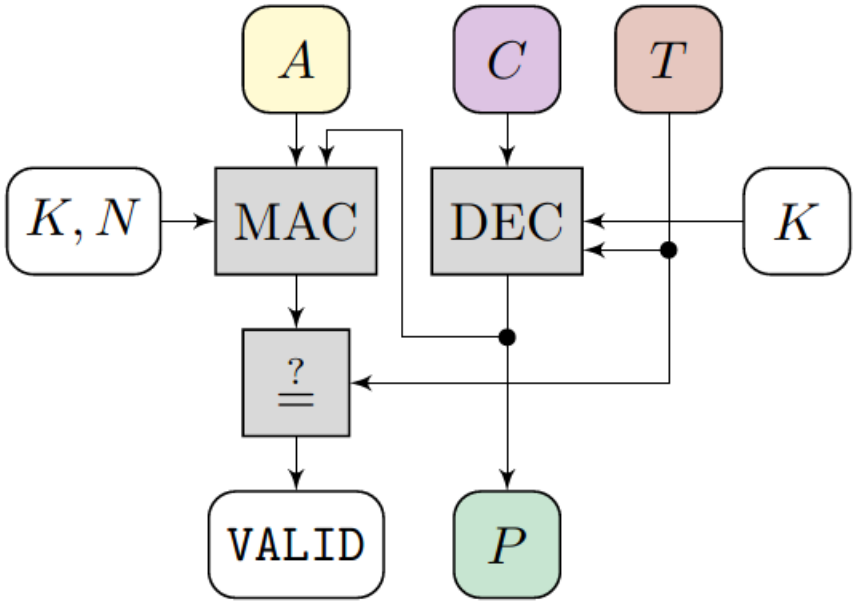
Synthetic Initialization Vector

Summary Flow *New*

Authenticated Encryption (Alice)

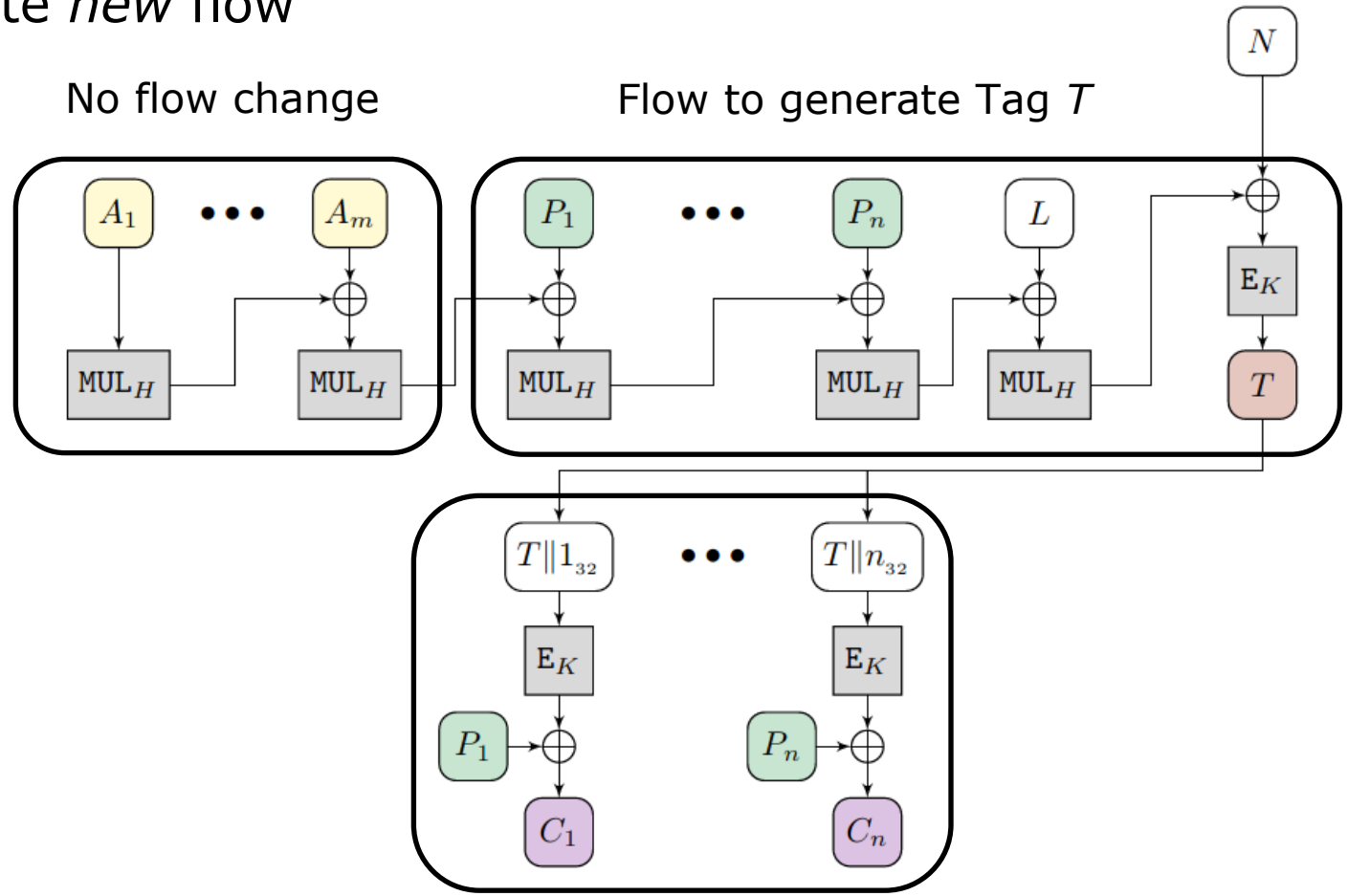


Authenticated Decryption (Bob)



AES-GCM-SIV Encryption [7]

Complete *new* flow



New flow using Tag *T* instead of Nonce *N*

AES-GCM-SIV Encryption

- Assessment

  => reuse of existing HW accelerators
  -> change only on input parameters

- => low risk regarding new design / HW flaws

Summary

- New SIV based AEAD evaluation is finished @ Infineon
- Implementation was tested on latest AURIX controller
- Result: Security improvement comes at limited investment

Literature

[1] J. Viega and D. A. McGrew, "The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH." [Online]. Available: https://tools.ietf.org/html/rfc4543. [Accessed: 07-Apr-2017].

[2] H. Böck, A. Zauner, S. Devlin, J. Somorovsky, and P. Jovanovic, "Nonce-disrespecting Adversaries: Practical Forgery Attacks on GCM in TLS," in *Proceedings of the 10th USENIX Conference on Offensive Technologies*, Berkeley, CA, USA, 2016, pp. 15–25.

[3] A. Joux, "Authentication failures in NIST version of GCM," Jan. 2006.

[4] P. Rogaway and T. Shrimpton, "Deterministic Authenticated-Encryption: A Provable-Security Treatment of the Key-Wrap Problem," 221, 2006.

Thank You !