Building the Next Generation of Authenticated Encryption

Mihir Bellare, Shay Gueron, Viet Tung Hoang, Julia Len, Sanketh Menda, and Thomas Ristenpart

Authenticated Encryption
with Associated Data

Scheme **AEAD**

Key **K**

Nonce **N**

Associated data **A**

Message **M**

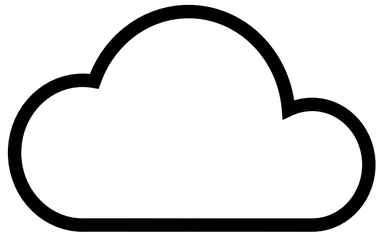
C ← **AEAD**.Enc(**K**, **N**, **A**, **M**)

N ‖ **A** ‖ **C**

Scheme **AEAD**

Key **K**

M ← **AEAD**.Dec(**K**, **N**, **A**, **C**)



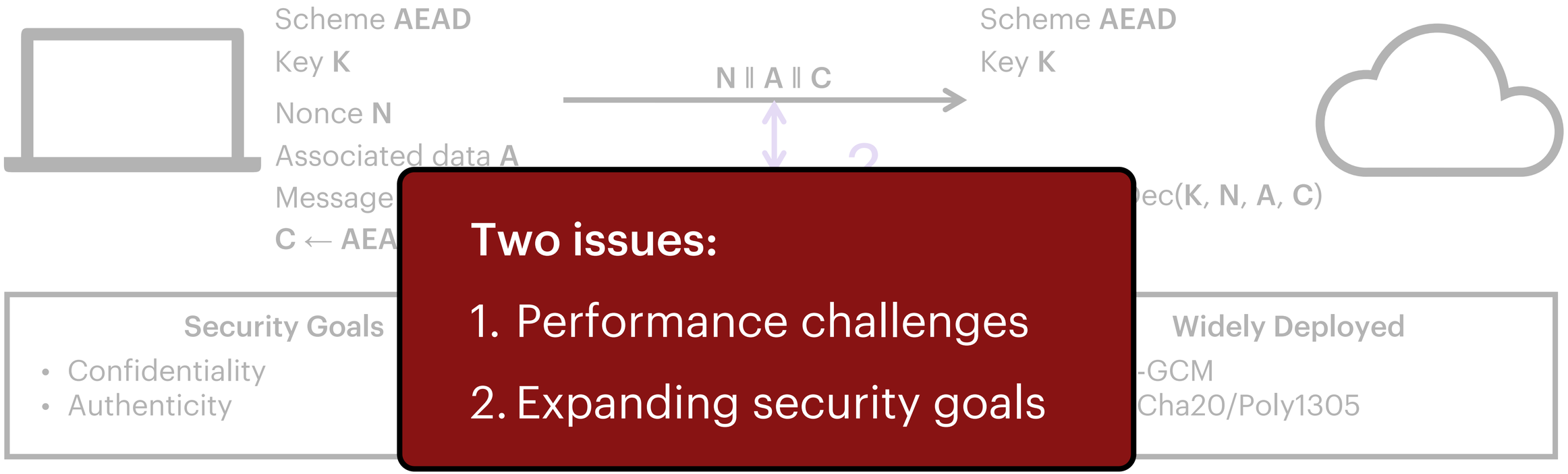
Security Goals

- Confidentiality
- Authenticity

Widely Deployed

- AES-GCM
- ChaCha20/Poly1305

Authenticated Encryption
with Associated Data

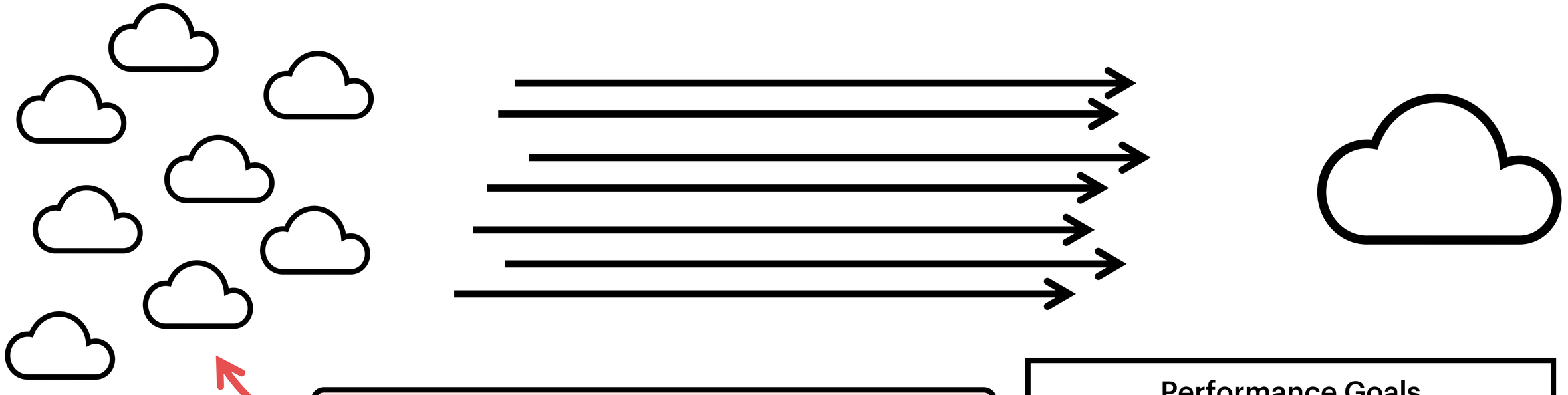


1 – Performance challenges

Performance Goals

- Streamable
- Fast with AES-NI hardware

1 – Performance challenges



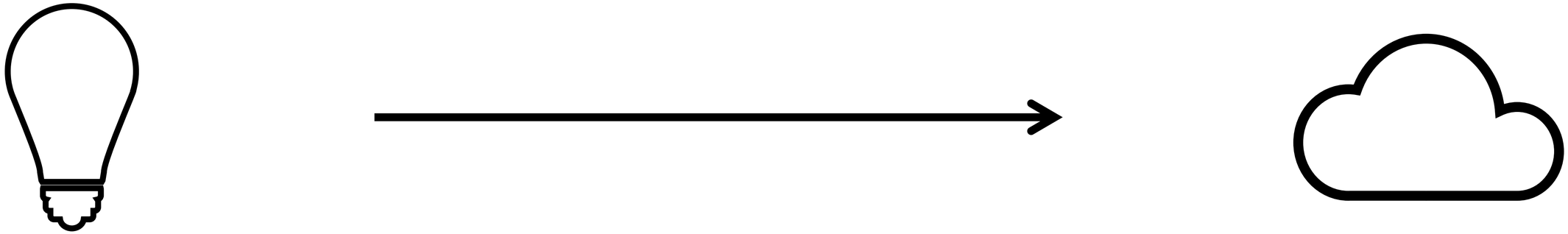
Scaling to modern workloads

- AWS: 2^{32} messages *in 2 seconds*
- AES-GCM can encrypt < 2^{32} messages per key (for random nonces)

Performance Goals

- Streamable
- Fast with AES-NI hardware
- **Scalability** (2^{96} encryptions per key for random nonces)

1 – Performance challenges



Performance on lightweight devices

- No AES instructions, so AES is too slow
- NIST Lightweight competition

Performance Goals

- Streamable
- Fast with AES-NI hardware
- **Scalability** (2^{96} encryptions per key for random nonces)
- **Fast on lightweight devices**

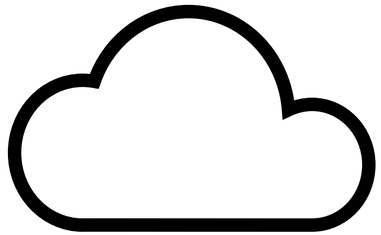
2 – Expanding security goals

Scheme **AEAD**
Key **K**
Nonce **N**
Associated data **A**
Message **M**
C ← **AEAD**.Enc(**K**, **N**, **A**, **M**)

N ‖ **A** ‖ **C**

Scheme **AEAD**
Key **K**

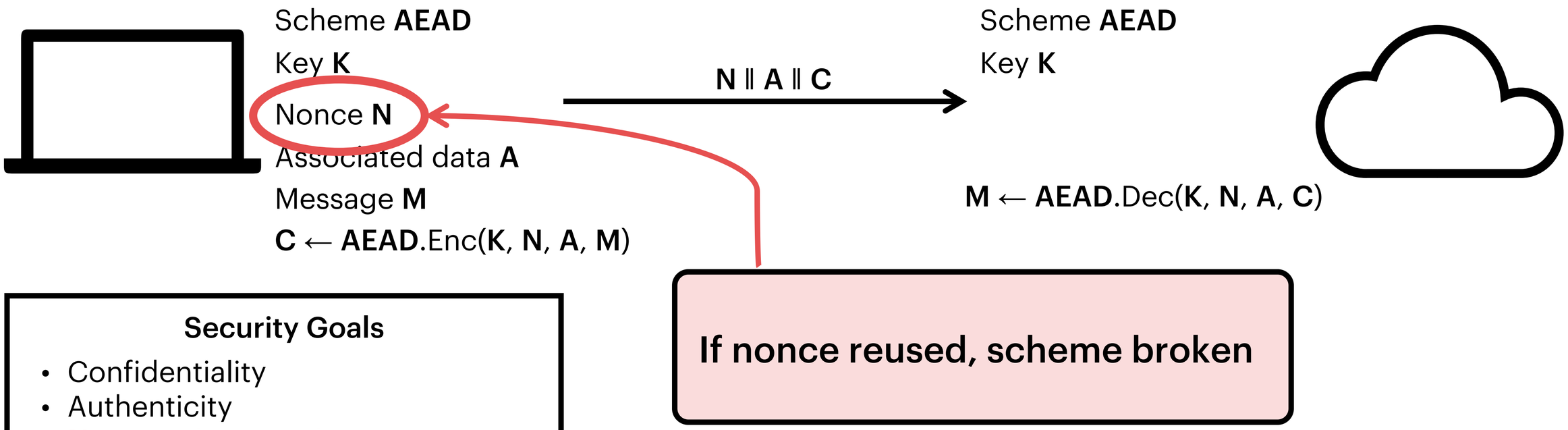
M ← **AEAD**.Dec(**K**, **N**, **A**, **C**)



Security Goals

- Confidentiality
- Authenticity

2 – Expanding security goals



Security Goals

- Confidentiality
- Authenticity
- Nonce-misuse resistance [RS EC06]

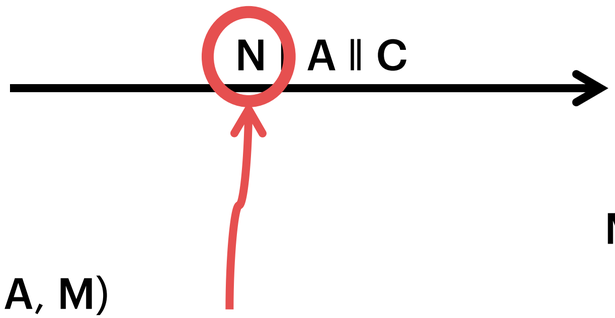
If nonce reused, scheme broken

Real world attacks:

- Inject malicious content into HTTPS sessions [BZDSJ WOOT'16]
- Extract keys from Samsung TrustZone [SRW USENIX Sec'22]

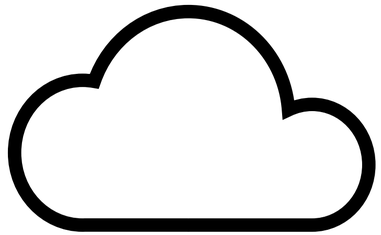
2 – Expanding security goals

Scheme **AEAD**
Key **K**
Nonce **N**
Associated data **A**
Message **M**
C ← **AEAD**.Enc(**K**, **N**, **A**, **M**)



Scheme **AEAD**
Key **K**

M ← **AEAD**.Dec(**K**, **N**, **A**, **C**)



Security Goals

- Confidentiality
- Authenticity
- Nonce-misuse resistance [RS EC06]
- Nonce hiding [BNT Crypto19]

Nonces are sent in the clear

Privacy leaks: [BNT Crypto'19]

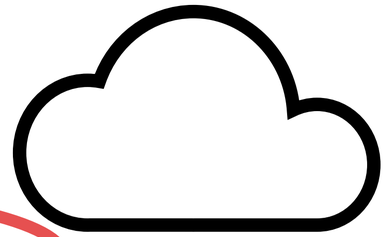
- Can reveal information about the session; e.g., counters.
- Can reveal information about the sender; e.g., machine identifiers
- Can be plain bad choices; e.g., hash of the message

2 – Expanding security goals

Scheme **AEAD**
Key **K**
Nonce **N**
Associated data **A**
Message **M**
C ← **AEAD**.Enc(**K**, **N**, **A**, **M**)

N ‖ **A** ‖ **C**

Scheme **AEAD**
Key **K**



M ← **AEAD**.Dec(**K**, **N**, **A**, **C**)

Decryption may succeed under different contexts (K, N, A)

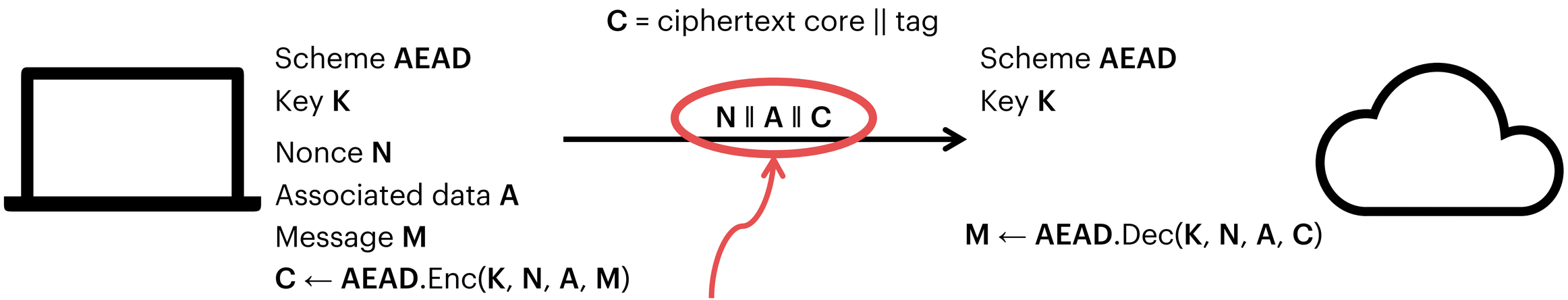
Security Goals

- Confidentiality
- Authenticity
- **Nonce-misuse resistance** [RS EC06]
- **Nonce hiding** [BNT Crypto19]
- **Context commitment** [BH EC22]

Real world attacks:

- Abuse reporting in Facebook Messenger [DGRW CRYPTO'18]
- Envelope encryption in AWS encryption SDK [ADGKLS USENIX Sec'22]

2 – Expanding security goals



Output should be not much longer than the message

Security Goals

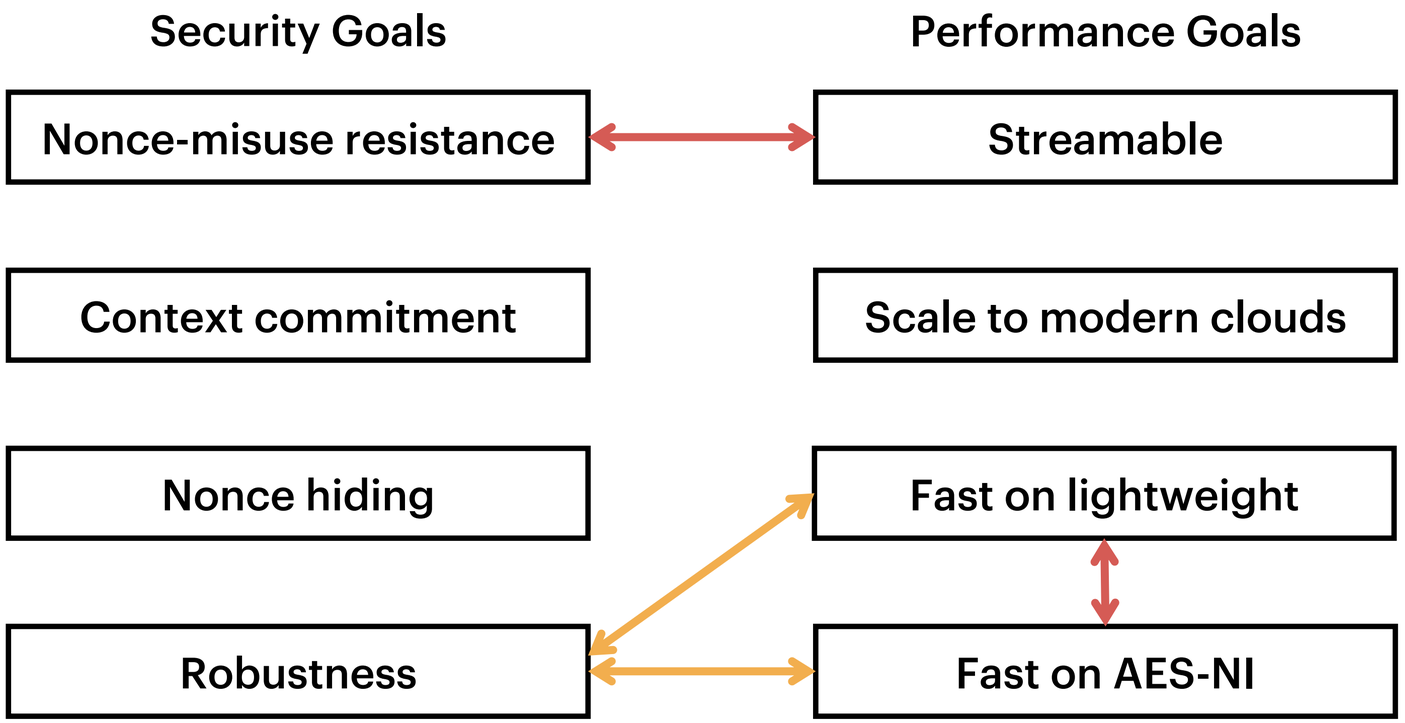
- Confidentiality
- Authenticity
- **Nonce-misuse resistance** [RS EC06]
- **Nonce hiding** [BNT Crypto19]
- **Context commitment** [BH EC22]
- **Robustness** [HKR EC15]

Real world interest:

- Android encrypts file contents with XTS or Adiantum [Android 14]
- NIST wants to standardize an “accordion cipher mode.”

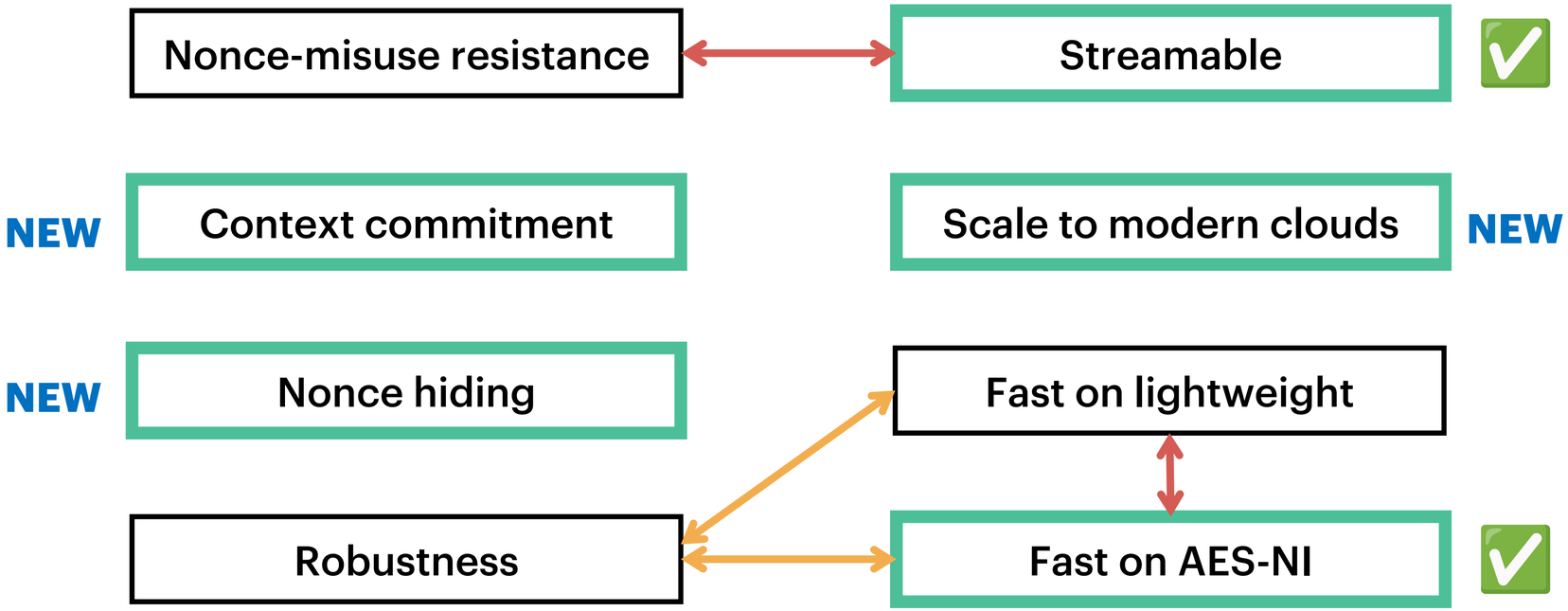
One scheme with all properties? No

Performance and security always in some level of tension!



One scheme with all properties? No

Performance and security always in some level of tension!

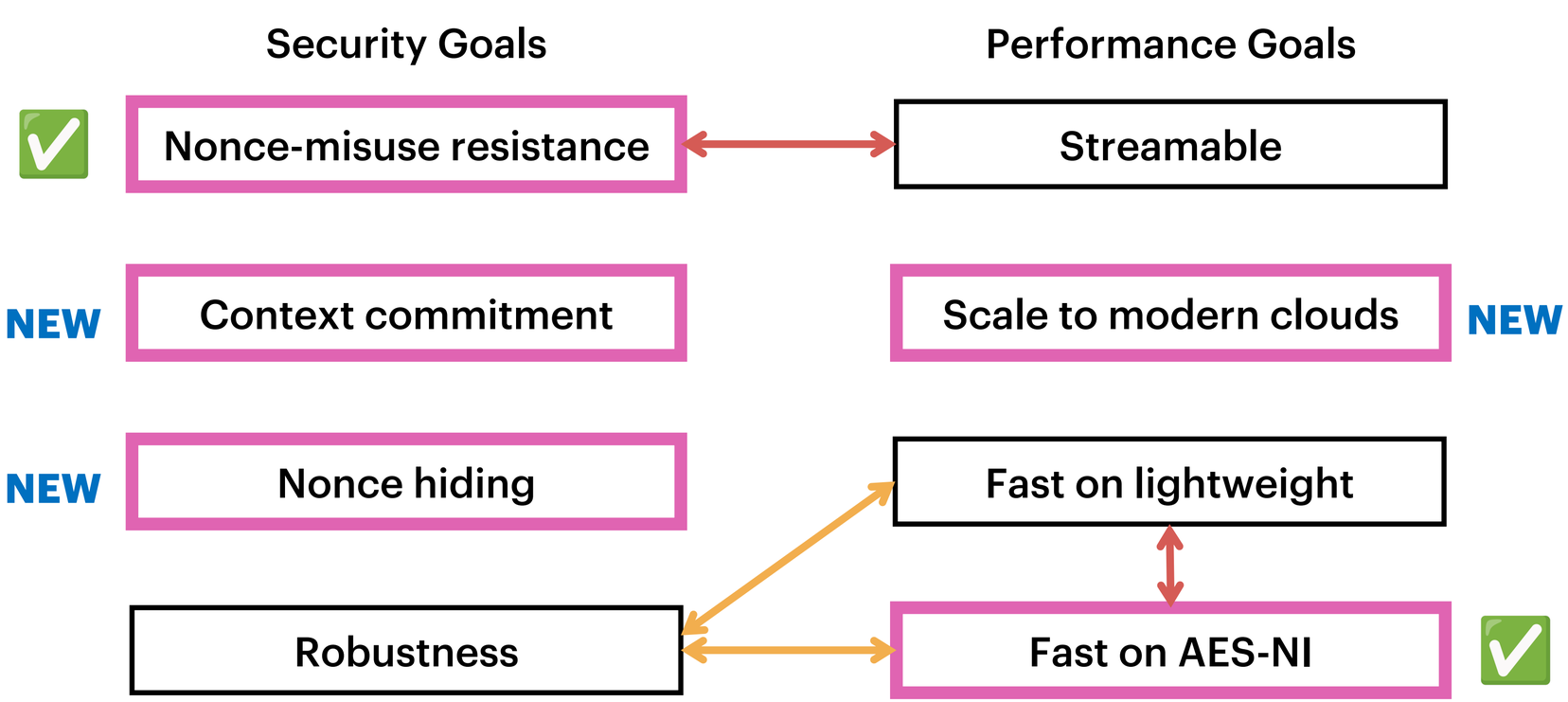


Setting nickname:

Streamable AES-GCM

One scheme with all properties? No

Performance and security always in some level of tension!



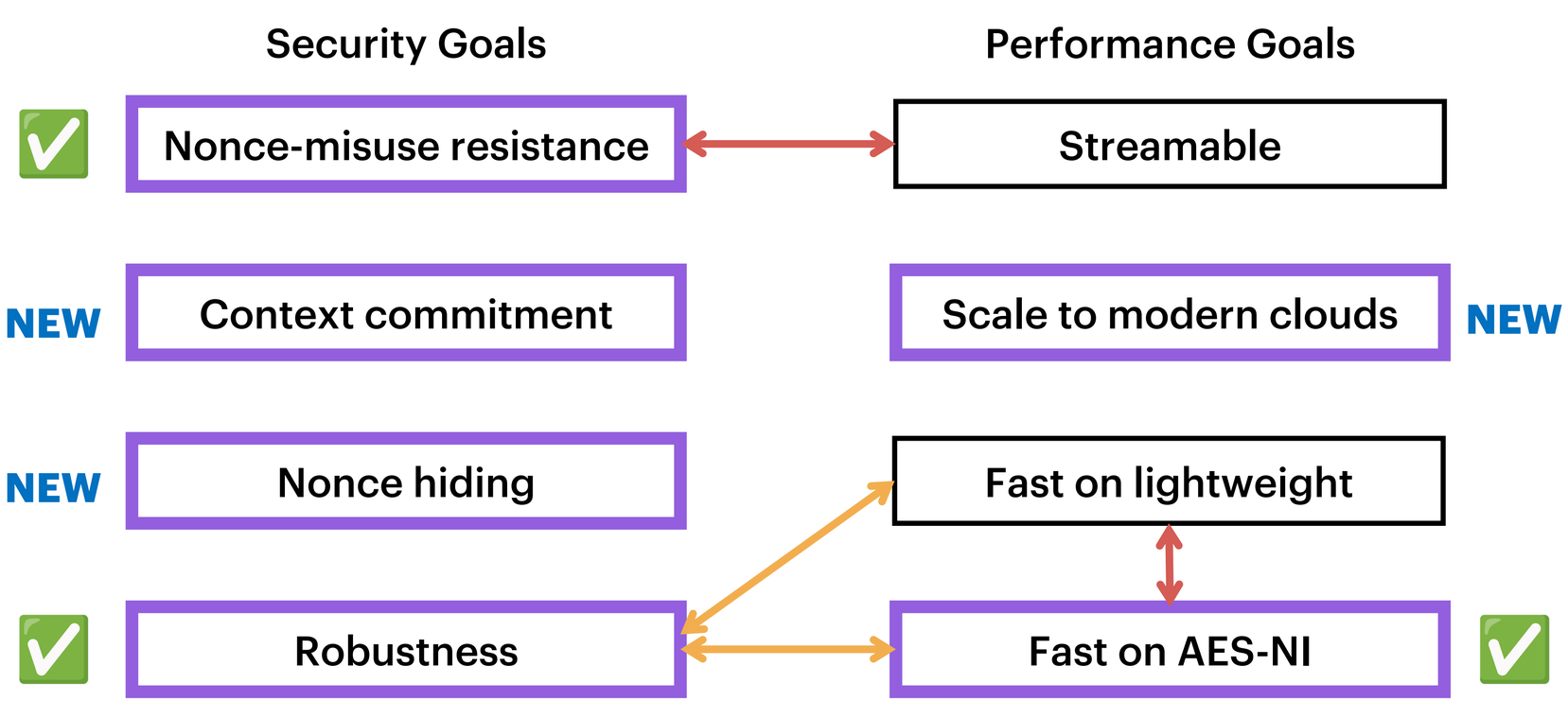
Setting nickname:

Streamable

Nonce MR AES-GCM-SIV

One scheme with all properties? No

Performance and security always in some level of tension!



Setting nickname:

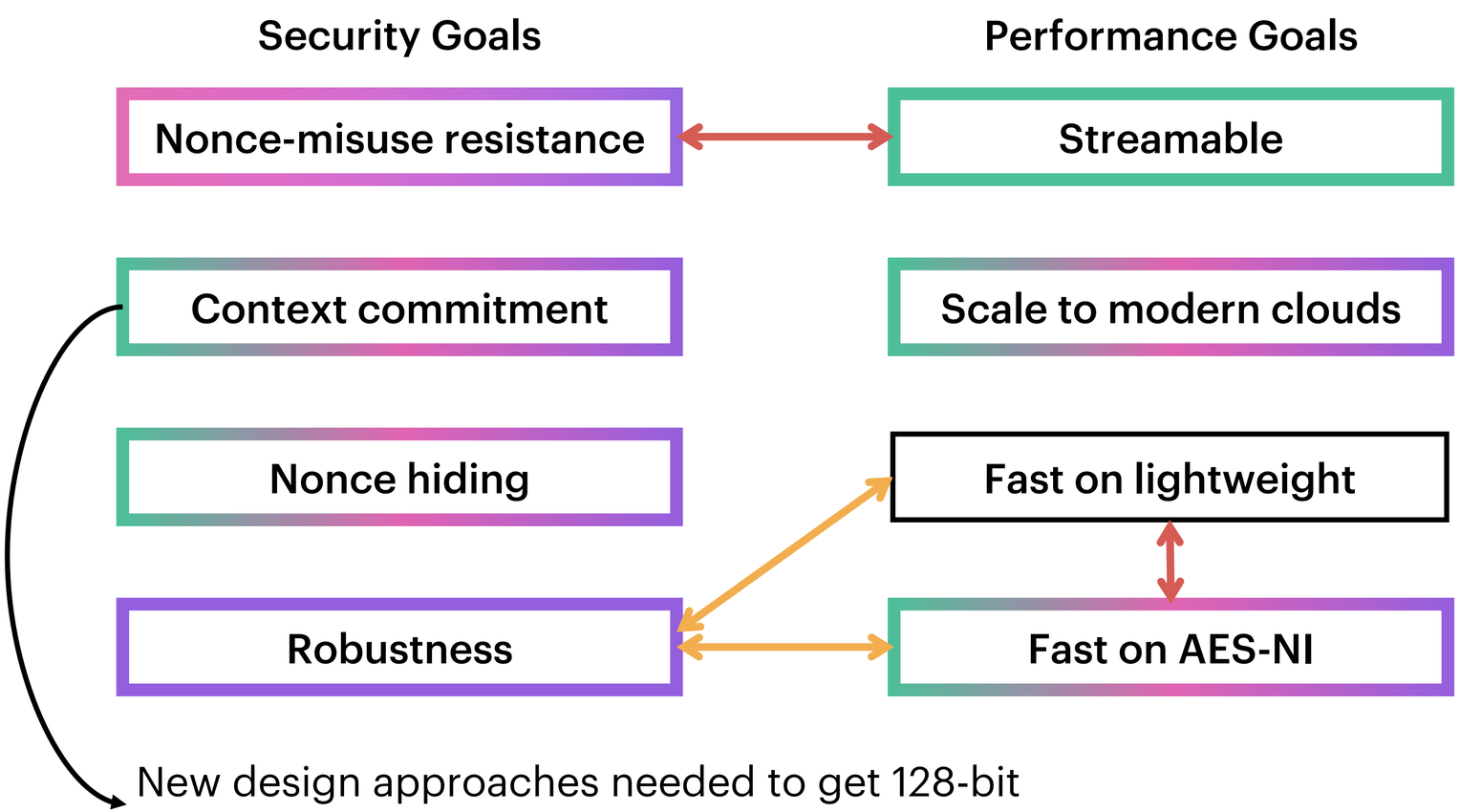
Streamable

Nonce MR

Robust AES-AEZ

One scheme with all properties? No

Performance and security always in some level of tension!



Setting nickname:

Streamable

Nonce MR

Robust

Streamable (lightweight)

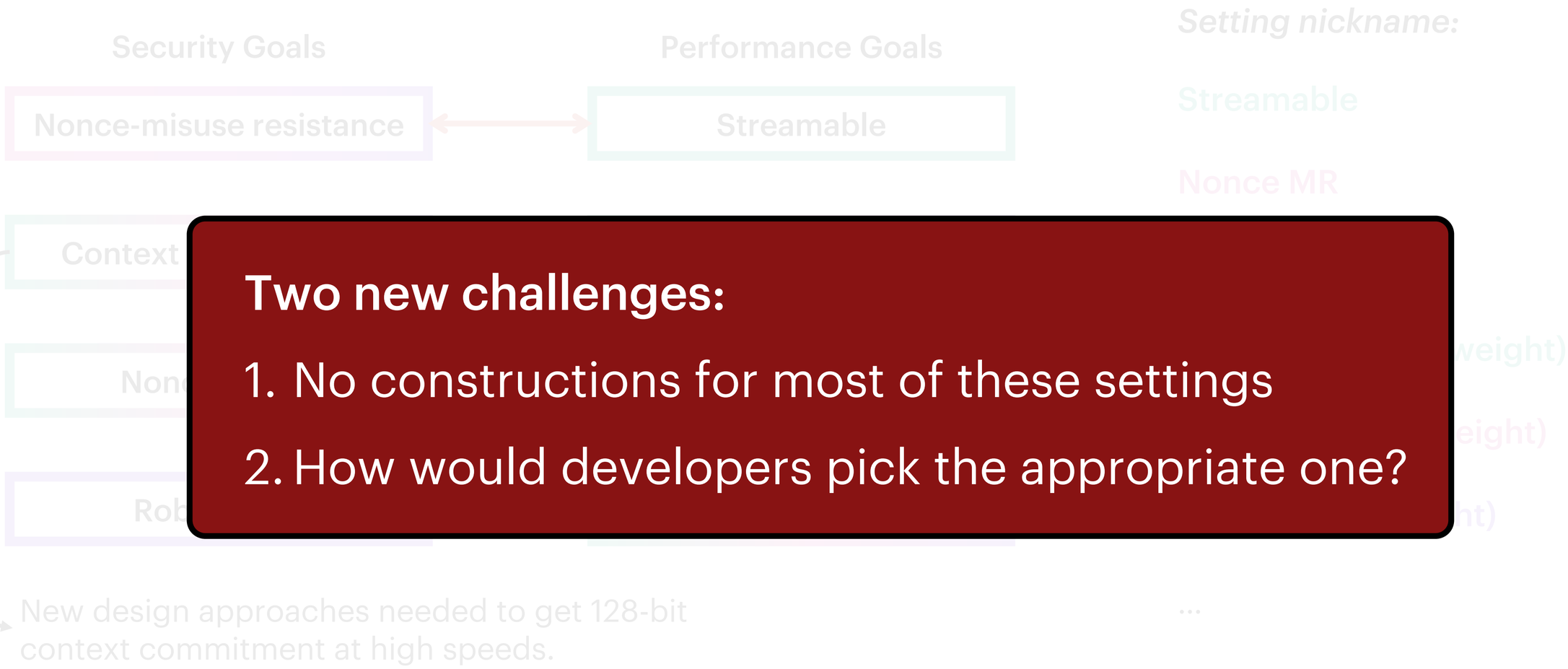
Nonce MR (lightweight)

Robust (lightweight)

…

One scheme with all properties? No

Performance and security always in some level of tension!



Two new challenges:

1. No constructions for most of these settings
2. How would developers pick the appropriate one?

Current AEADs don’t meet all our goals

We can’t have one do-everything AEAD

We need many new AEAD schemes…

…and an easy way to pick the right one

Our vision for next generation AEAD

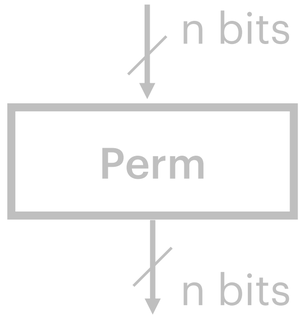
New suite of AEAD schemes targeting streamable, nonce-MR, and robust AEAD settings

- Context committing
- Nonce-hiding supported

GCH OCH CIV HCTR3

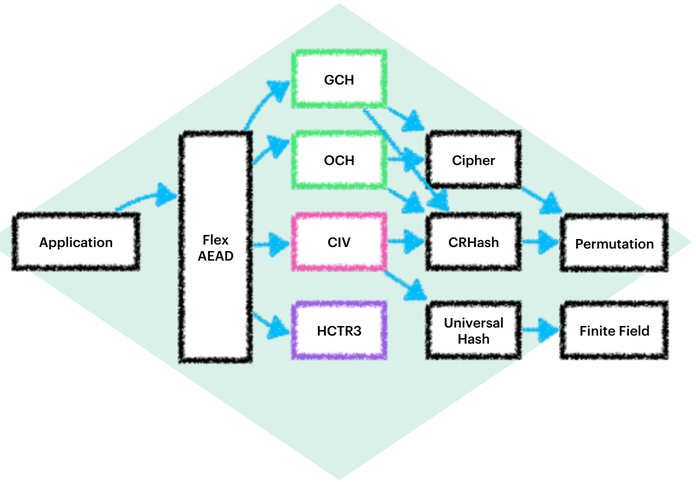
AEAD as **modes of operation of cryptographic permutation(s)**

- Builds off permutation-based cryptography [Keccak Team]
- Many great permutations, some leveraging AES-NI
- Wider block sizes than AES-128
- Perms good for both lightweight and desktop

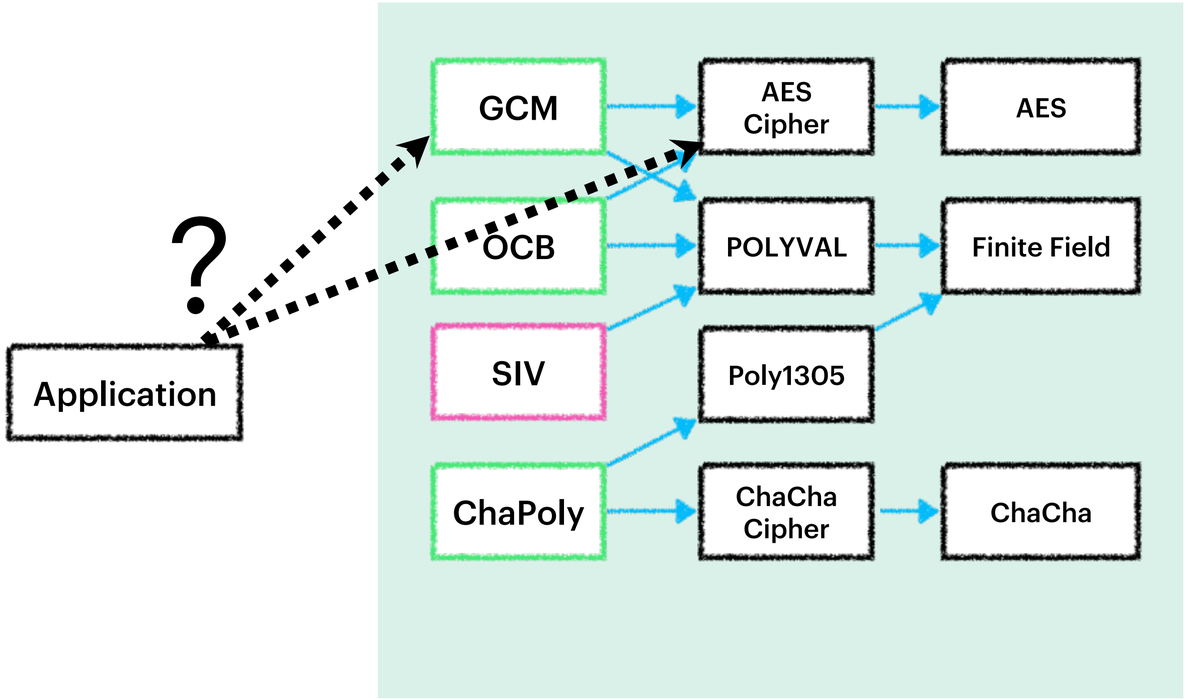


Keccak (n = 1600, 800,...)
Ascon (n = 320)
Simpira (n = 256, 512,...)
Areion (n = 256, 512)
...

Flexible AEAD abstraction that combines different modes to make it easier to use them securely



Current approach leads to complex landscape



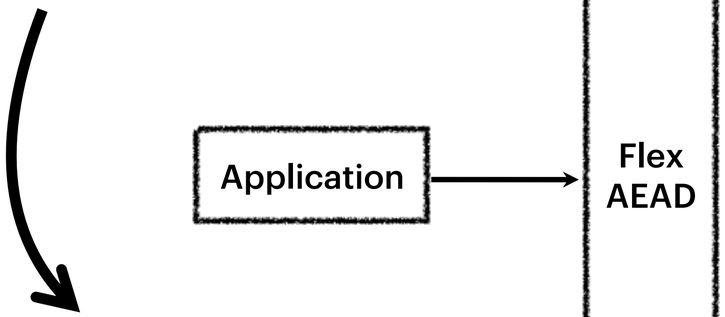
Increasing complexity, lots of components

Developers have to pick schemes and parameters

If decrypt with wrong scheme or key, no security guarantees

A new approach: Flexible AEAD

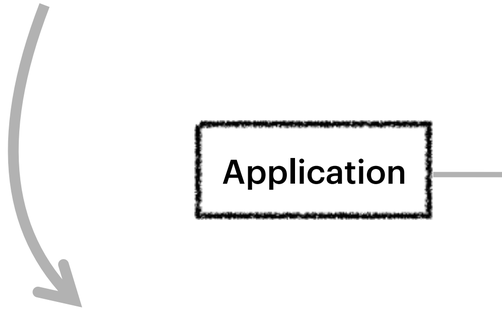
Developers pick configuration of properties. Safe defaults



config	
Misuse resistance	TRUE
Context commitment	TRUE
Nonce hiding	TRUE
AES-NI support	FALSE
.	.
.	.
.	.

A new approach: Flexible AEAD

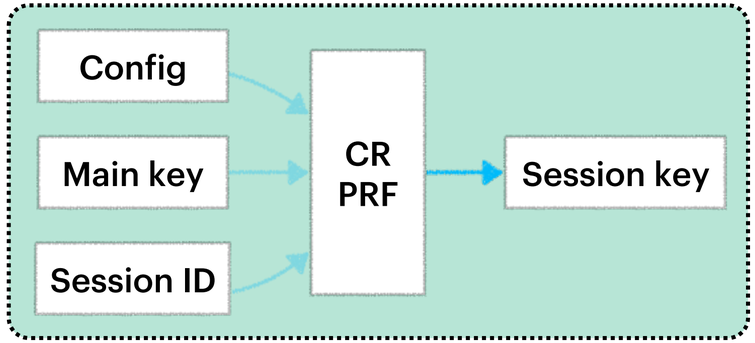
Developers pick configuration of properties. Safe defaults



config	
Misuse resistance	TRUE
Context commitment	TRUE
Nonce hiding	TRUE
AES-NI support	FALSE
.	.
.	.
.	.

Flex chooses scheme based on configuration

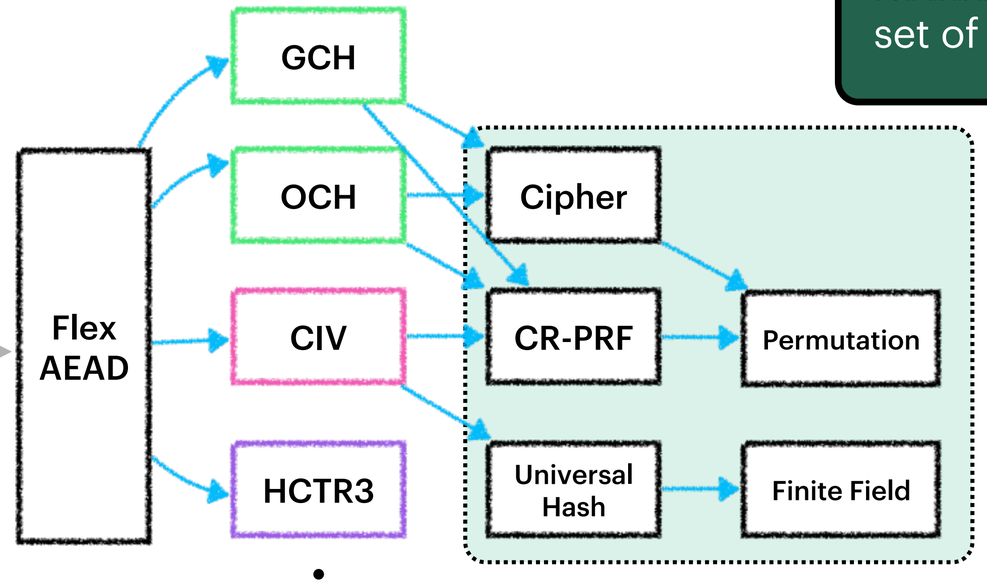
Session key derived from configuration avoids misuse [Tink]



A new approach: Flexible AEAD

Developers pick configuration of properties. Safe defaults

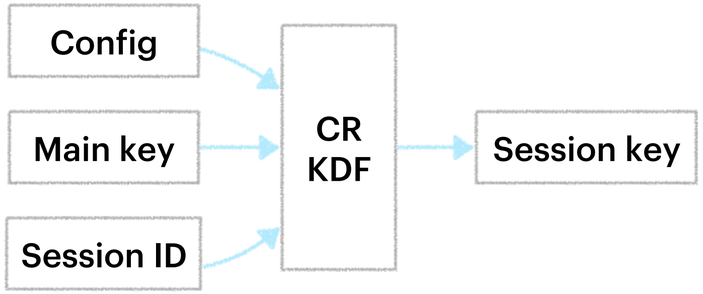
config	
Misuse resistance	TRUE
Context commitment	TRUE
Nonce hiding	TRUE
AES-NI support	FALSE
.	.
.	.
.	.



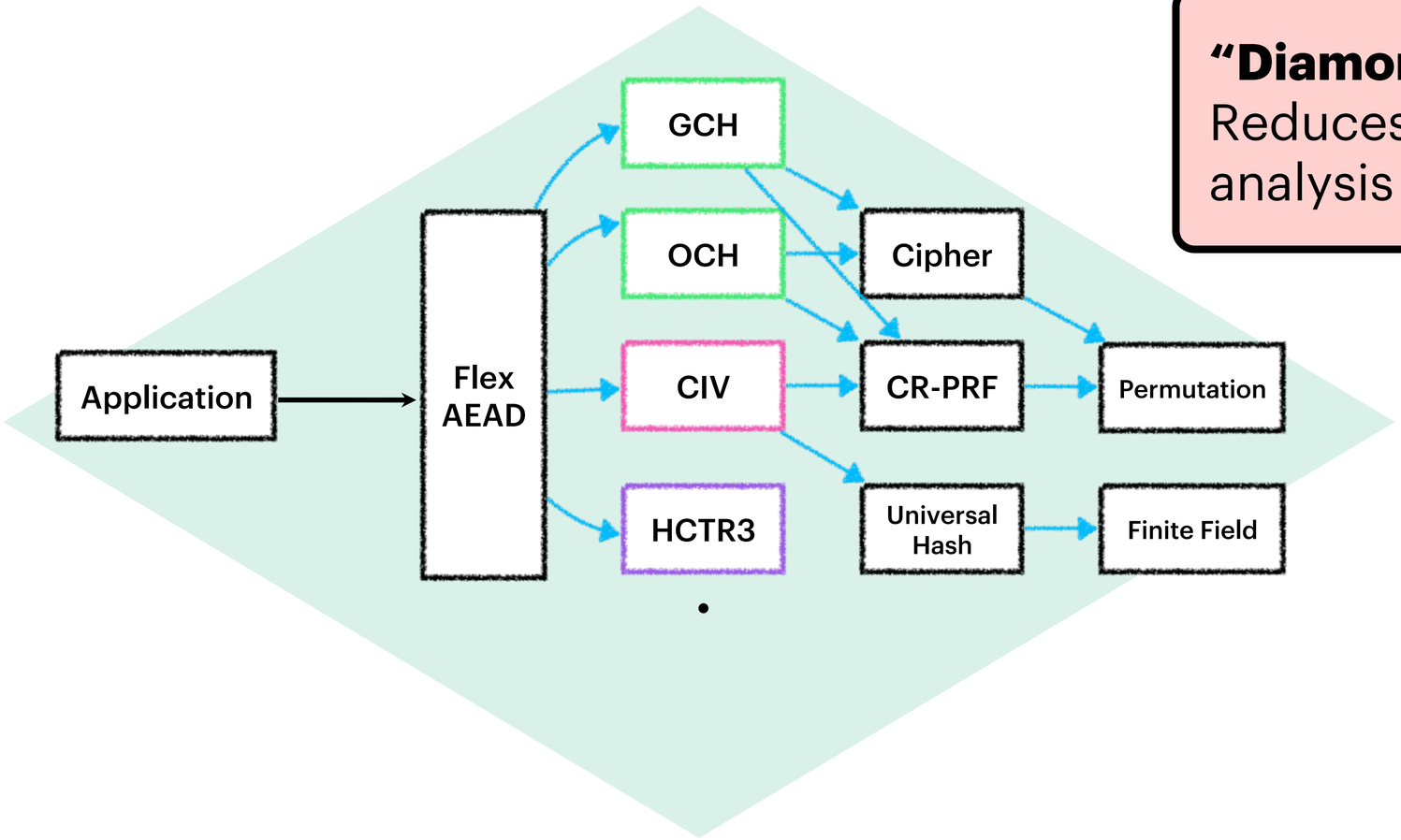
Reduced complexity: small set of underlying components

Flex chooses scheme based on configuration

Session key derived from configuration avoids misuse [Tink]

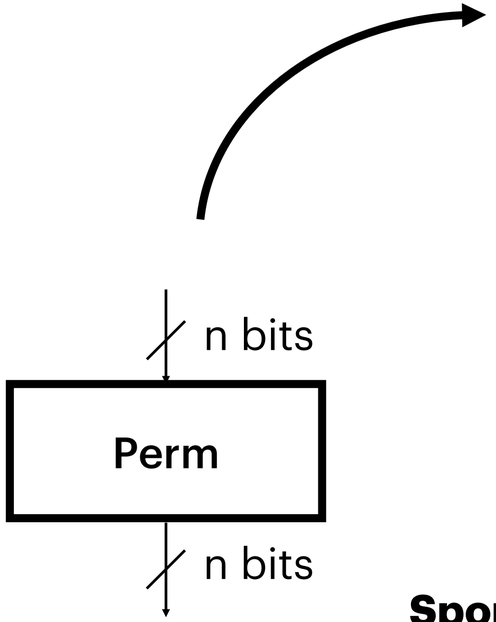


A new approach: Flexible AEAD

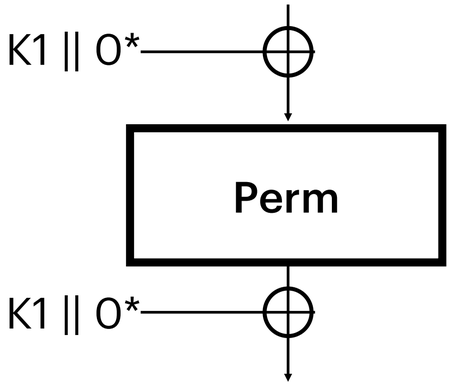


“Diamond strategy”:
Reduces implementation and analysis complexity

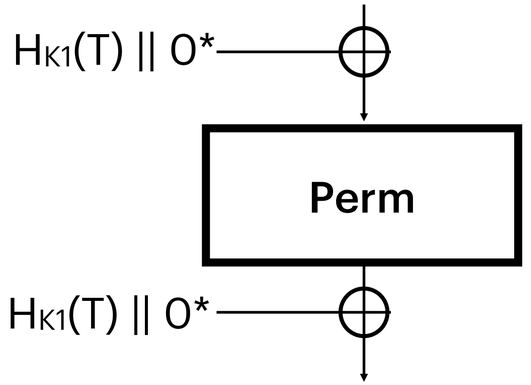
Underlying cryptographic components



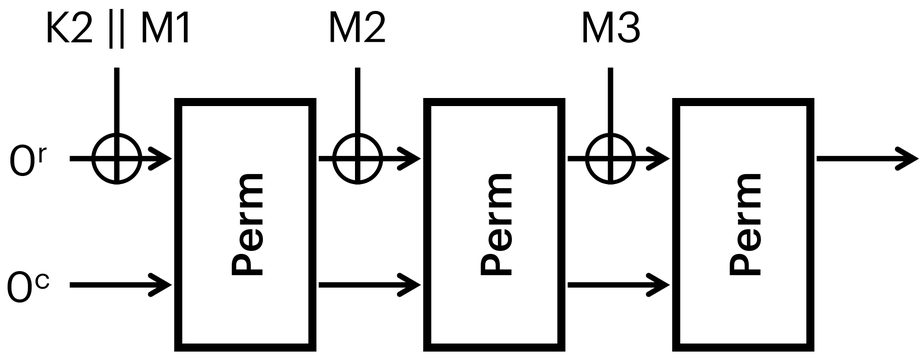
Even-Mansour blockcipher



Tweakable EM [CLS 2015]



Sponge-based CR-PRF:

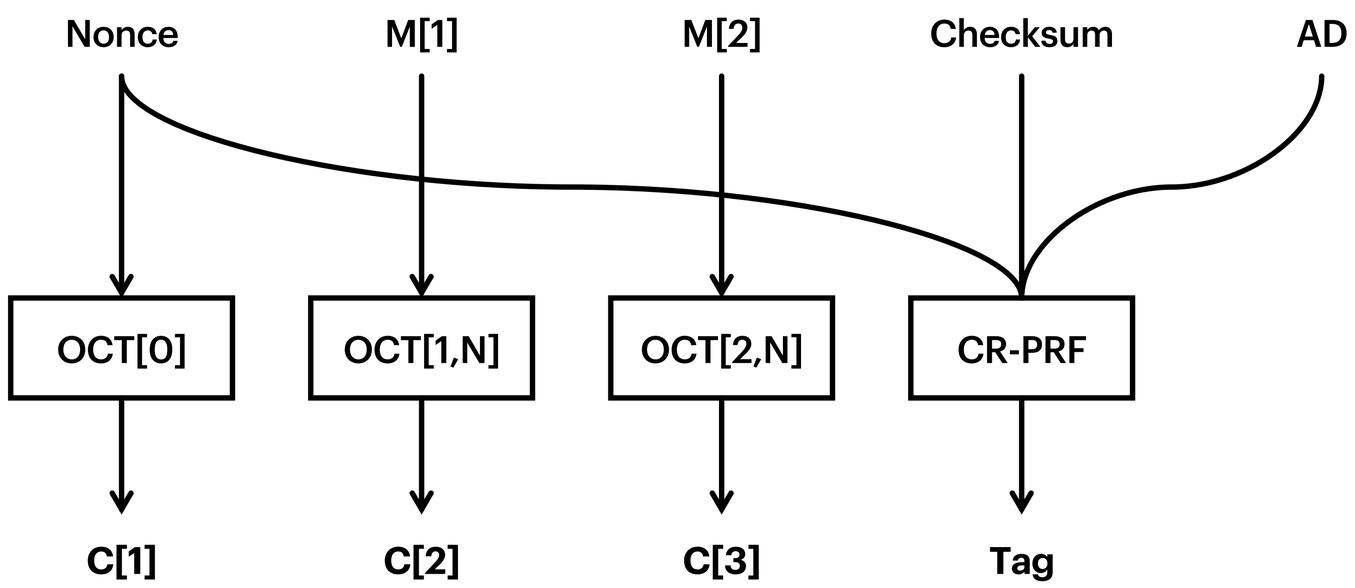


OCT tweakable blockcipher

- Specially designed, fast almost XOR universal H
- Formal security analysis

OCH: Committing OCB3-inspired AEAD

"OCB with Hashing"



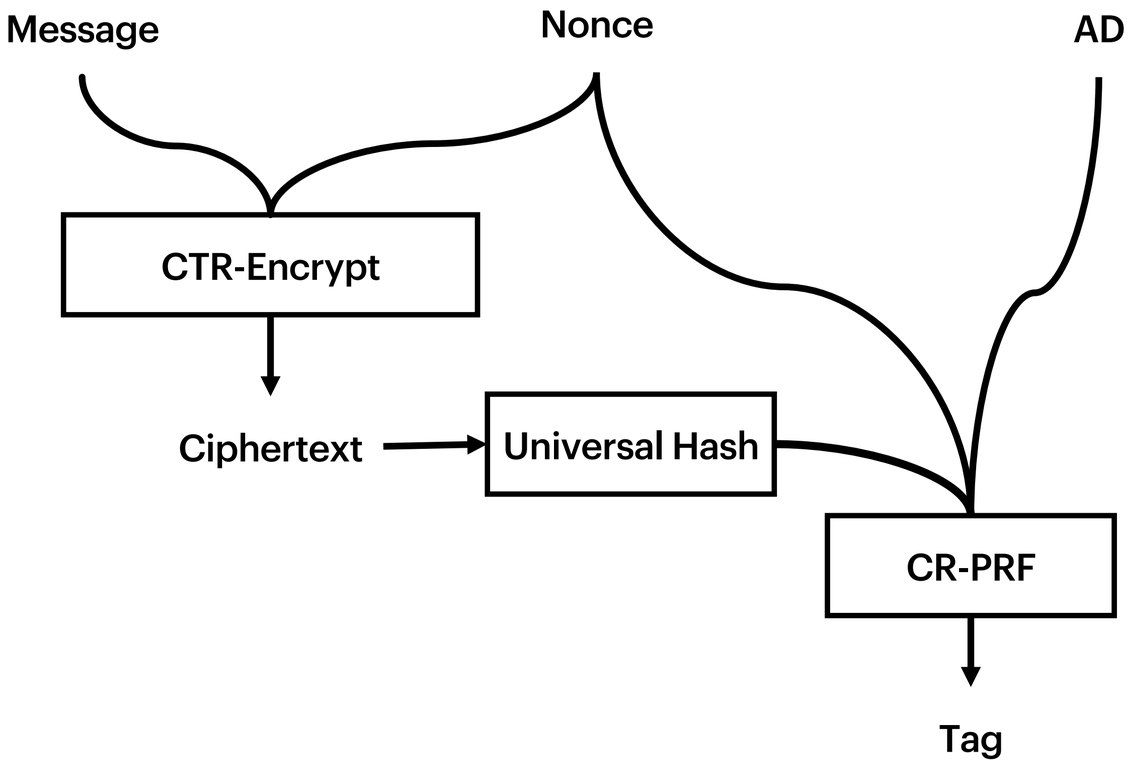
Simplified view:
OCT used in OCB3-like mode

Context committing
and nonce hiding

Fast, streamable scheme

GCH: Drop-in for GCM

"GCM with Hashing"



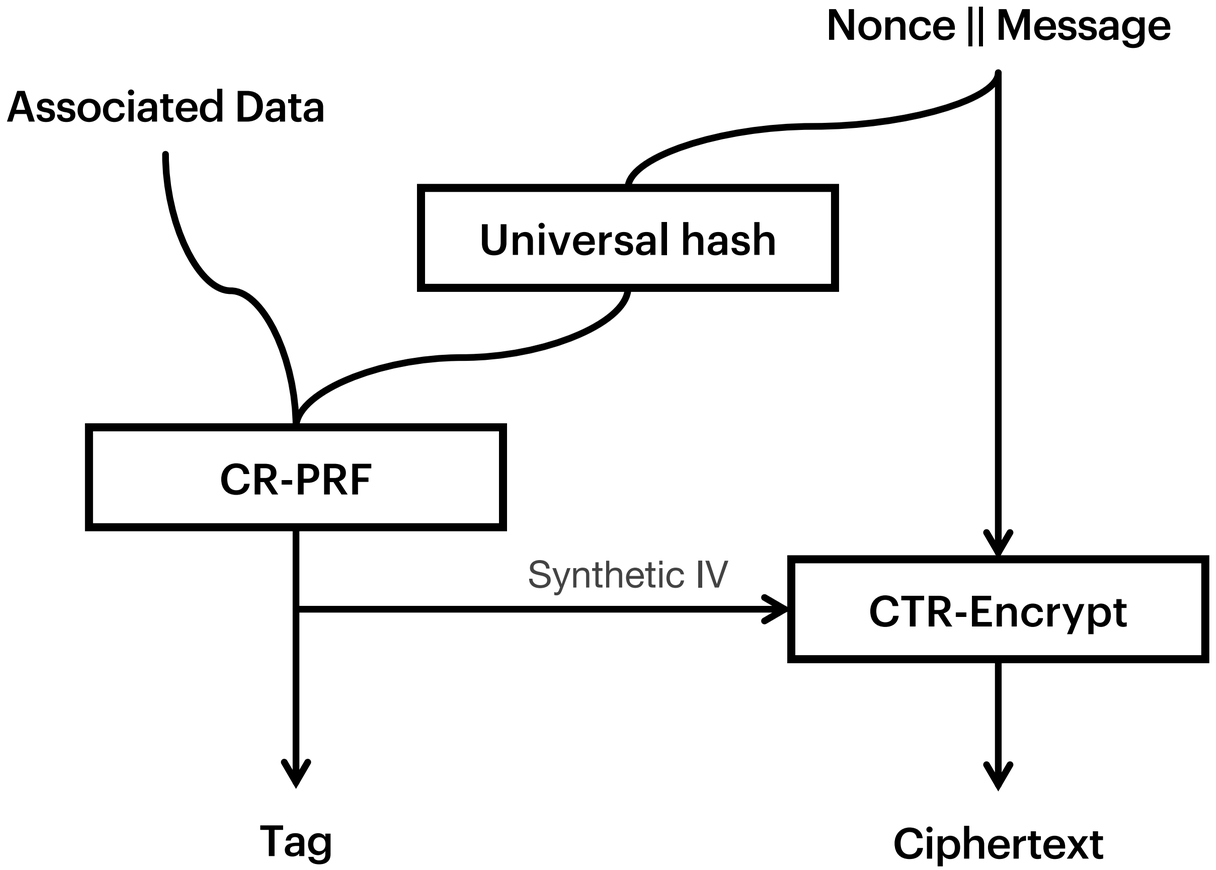
CTR mode using Even-Mansour: key stream precomputable

Can leverage AES-NI and PCLMULQDQ-NI pipelining

Fast, streamable scheme

CIV: Committing nonce-misuse resistance

"Committing SIV"



Context committing
and nonce-misuse resistance

Can leverage AES-NI and
PCLMULQDQ-NI

Fast, nonce-MR scheme

Summary: our vision for next generation AEAD

New suite of permutation-based AEAD schemes targeting streamable, nonce-MR, and robust AEAD settings

- Context committing
- Nonce-hiding supported
- Performant

Flexible AEAD abstraction that combines different modes to make it easier to use them securely

Please reach out: snkth.com