Committing Security of Authenticated Encryption

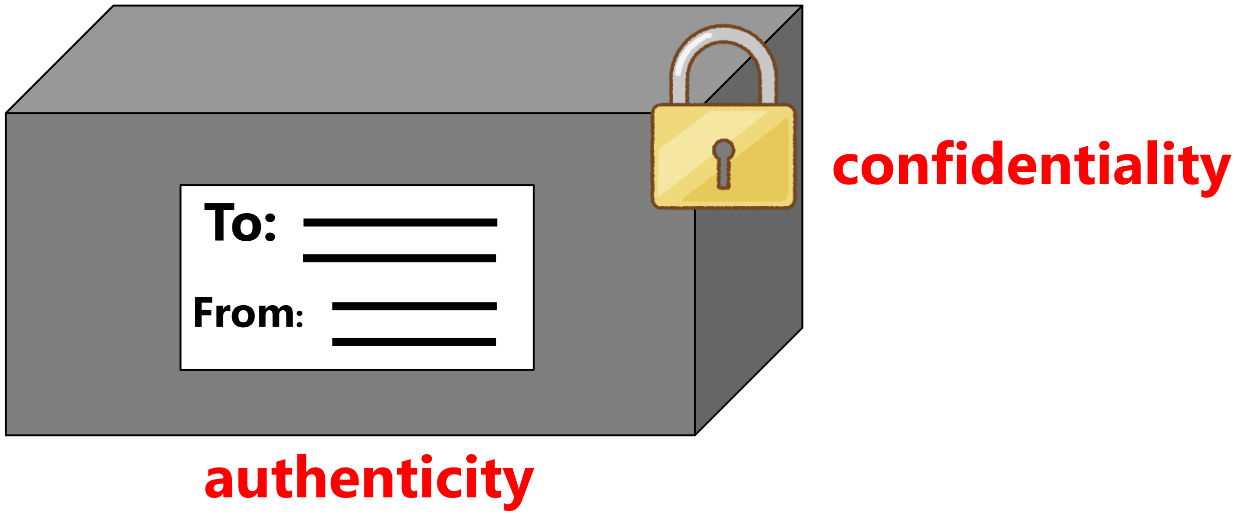
Yu Sasaki
NTT Social Informatics Laboratories, NIST Associate
2024.12.16 ASK2024 @ TCG CREST Kolkata

AEAD: Authenticated Encryption with Associated Data

Security for Communications

Two important features for secure communications:

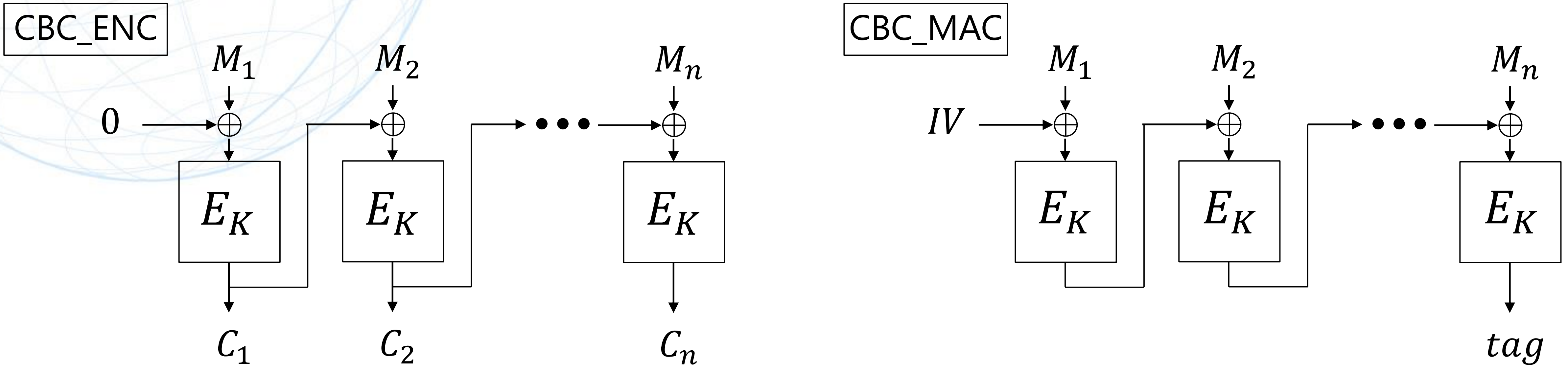
- **Confidentiality:** ensure that only legitimate users can read the data
  - Achieved by enciphering a plaintext to a ciphertext
- **Authenticity:** ensure that the data is not modified
  - Achieved by generating message authentication code (MAC)



Security for Communications

In early days, ciphers and MACs were developed independently.

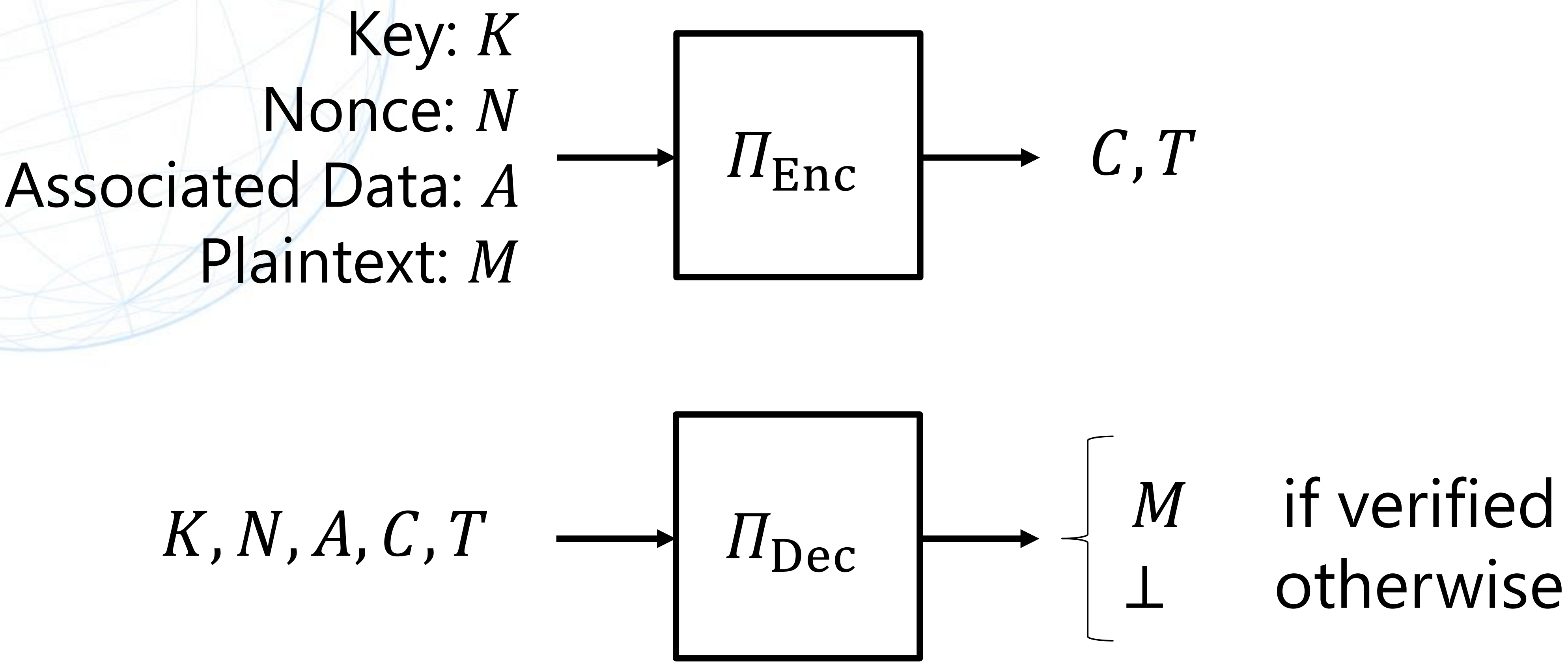
- Vulnerabilities could emerge by combining two: e.g. CBC_ENC + CBC_MAC
- Inefficient by computing ENC and MAC from scratch



Now a days, designing authenticated encryption with associated data (AEAD) is more popular to overcome those issues.

Internationally standardized: GCM, CCM, OCB, GCM-SIV, AEGIS, ASCON

Nonce-based AEAD Syntax



Conventional Security of AEAD (Intuitive)

NTT

- Security is considered for a single user with a single key

  Adversaries can interact only a single user.
  - **Privacy**: encrypted messages cannot be distinguished from a random string
  - **Integrity**: illegitimate uses cannot generate ciphertexts that pass verification

- In general, multiple users with different keys are connected to a single service.

  Adversaries can interact only a single user.

  Still, the considered security is the same: privacy and integrity.

Committing Security and Its Impact (CMT Security)

Pioneering Work: Key Robustness [FOR17]

Toward a theoretically ideal AEAD, the **key robustness**, later called "**key commitment**," was studied with several examples in mind.

Intuition:

- *Any given ciphertext would only be valid for a single secret key.*

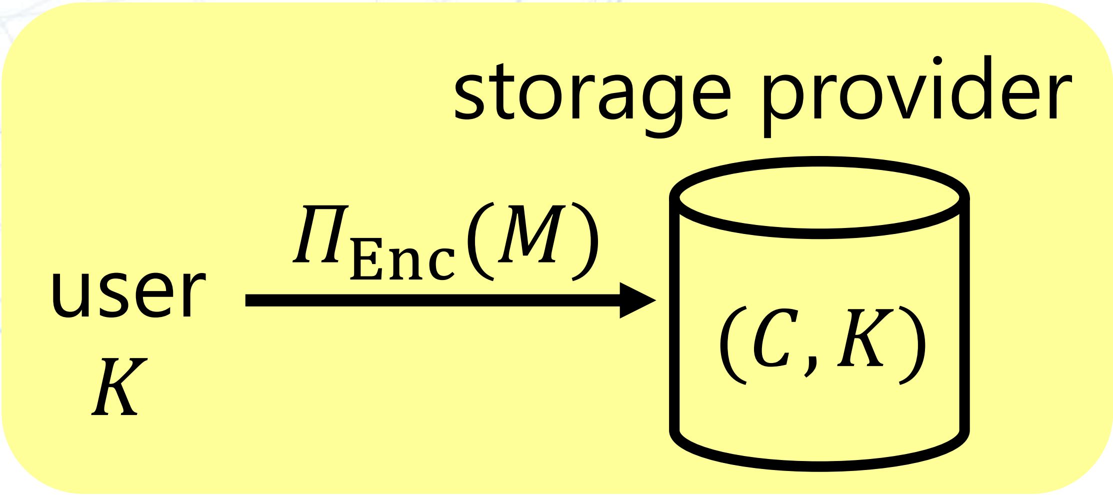
or

- *It must be hard to find two distinct keys reaching the same ciphertext.*

Key robustness is not covered by the conventional security notions.

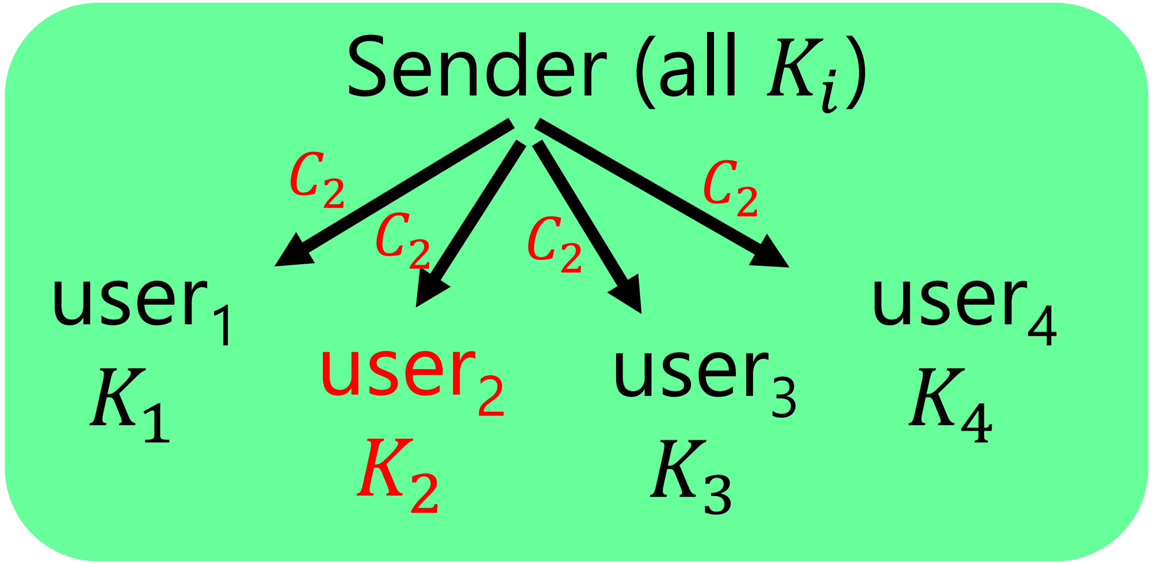
Relevance of Key Robustness [FOR17]

Ex.1 Storage Authenticity



If a malicious provider replaces K with K', verification must fail.

Ex.2 Anonymous Comm.



- A sender encrypts M for user$_2$ with K_2 to generate C_2.
- C_2 is broadcasted to all users.
- Other user$_i$ only learns $i \neq 2$.

Facebook's Message Franking

[GLR17] found Facebook's message franking is more relevant.

Message Franking Protocol

The goal is to resolve the following issue.

- Message franking is an end-to-end encrypted message system: intermediaries including service providers (Facebook) cannot see user's messages.
- When a user receives malicious message, the recipient should be able to report it to the service provider. But because of end-to-end confidentiality, the service provide cannot observe the actual message, and must rely on user's report.

Message Franking for Honest Alice (1/3)

NTT

- Allice chooses a key K_f.
- Generate a ciphertext C_1 and a tag C_2.

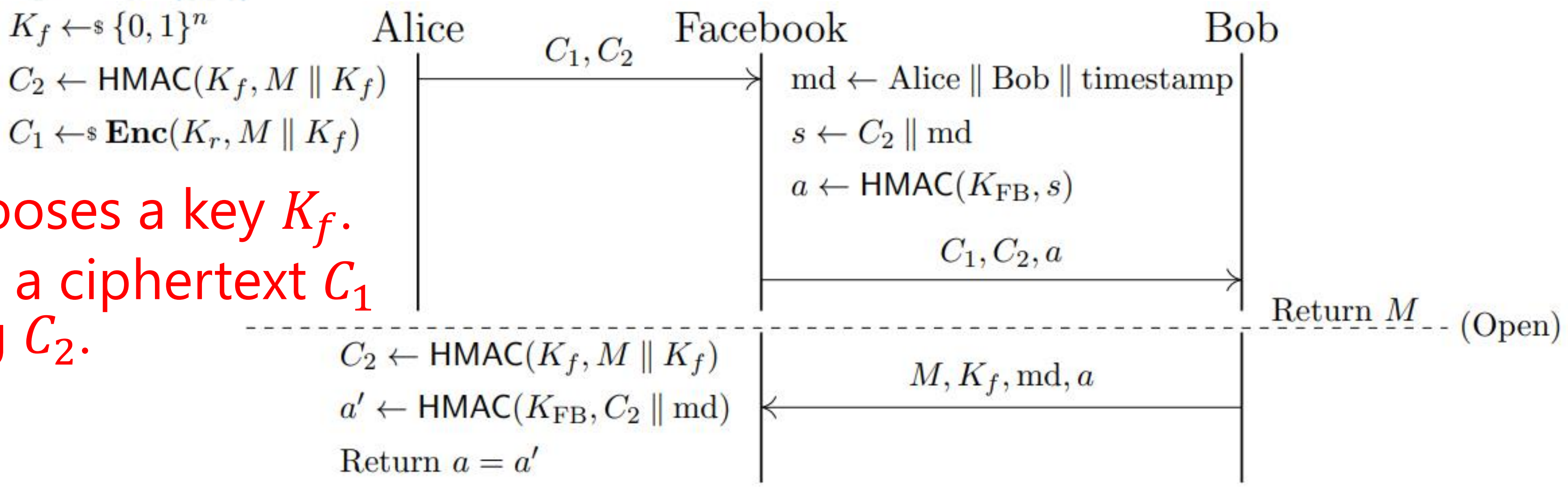


Figure 3: Facebook's message franking protocol [51]. The key K_r is a one-time-use symmetric key

(Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message Franking via Committing Authenticated Encryption")

Message Franking for Honest Alice (2/3)

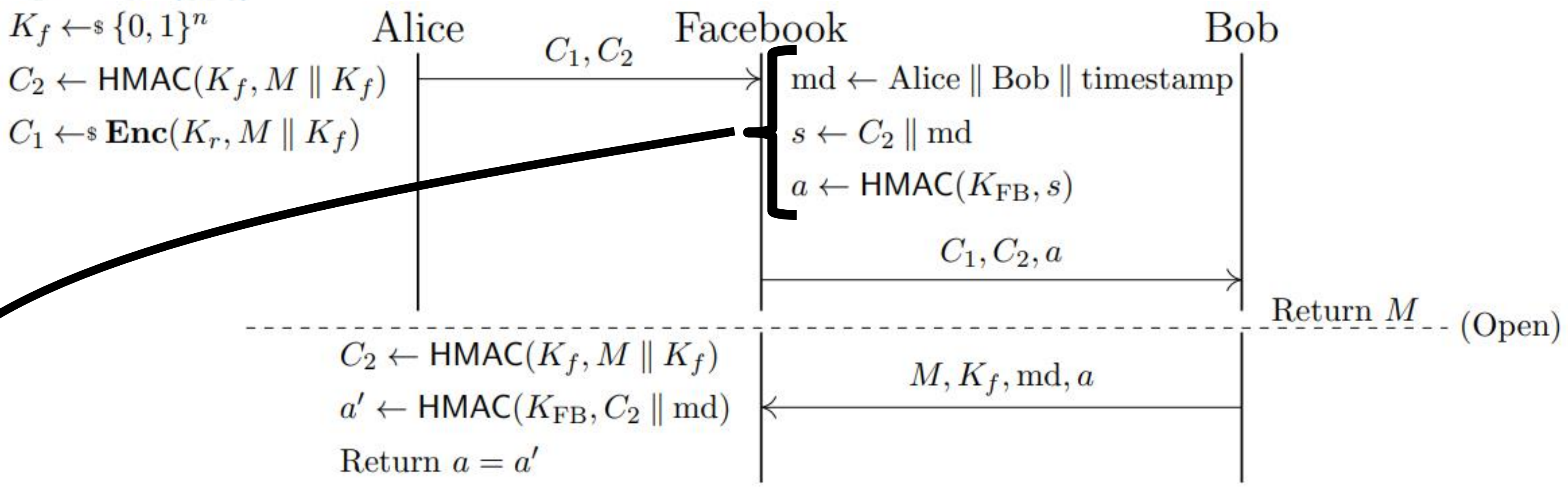


Figure 3: Facebook's message franking protocol [51]. The key K_r is a one-time-use symmetric key

(Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message Franking via Committing Authenticated Encryption")

- Facebook does not know K_f (end-to-end confidentiality).
- Facebook just authorizes metadata (communication players and timestamp) and the received tag value C_2 with Facebook's own key.

Message Franking for Honest Alice (3/3)

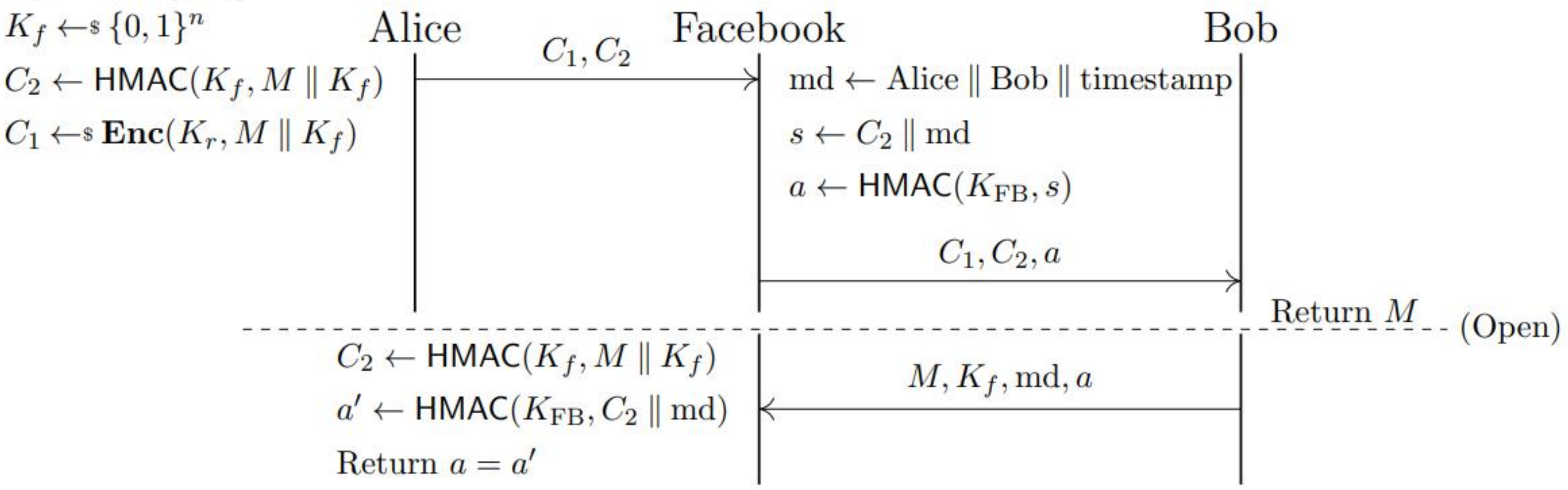


Figure 3: Facebook's message franking protocol [51]. The key K_r is a one-time-use symmetric key

(Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message Franking via Committing Authenticated Encryption")

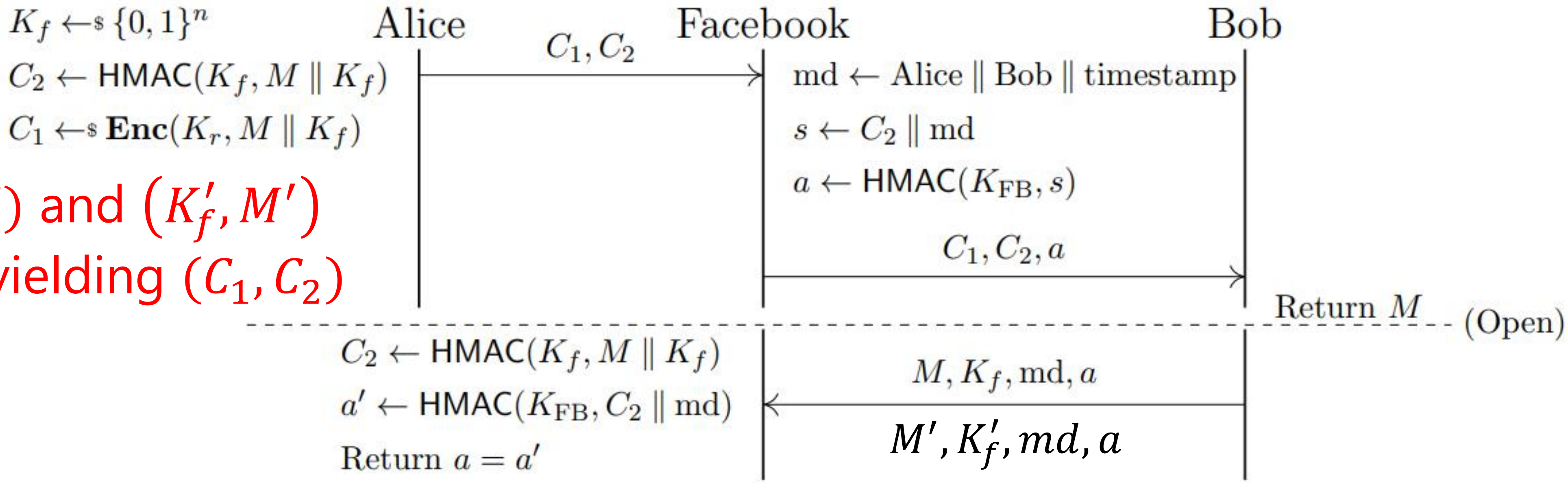
- If Bob finds M is malicious, Bob reports M, K_f, md, a to Facebook.
- Facebook checks authenticity of Bob's reports by computing C_2 then a.

Attack Scenario

- Alice wants to send a malicious message to Bob.
- Bob will report it to the service provider.
- Alice wants to avoid being punished even after the Bob's report.

Exploiting Lack of CMT Security by Alice (1/3)

NTT



(K_f, M) and (K_f', M') both yielding (C_1, C_2)

Figure 3: Facebook's message franking protocol [51]. The key K_r is a one-time-use symmetric key

(Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message Franking via Committing Authenticated Encryption")

- Alice can choose K_f and M.
- She prepares (K_f, M) and (K_f', M') both yielding (C_1, C_2), possibly M is chosen to be malicious and M' can be anything, e.g. random string.

Exploiting Lack of CMT Security by Alice (2/3)

NTT

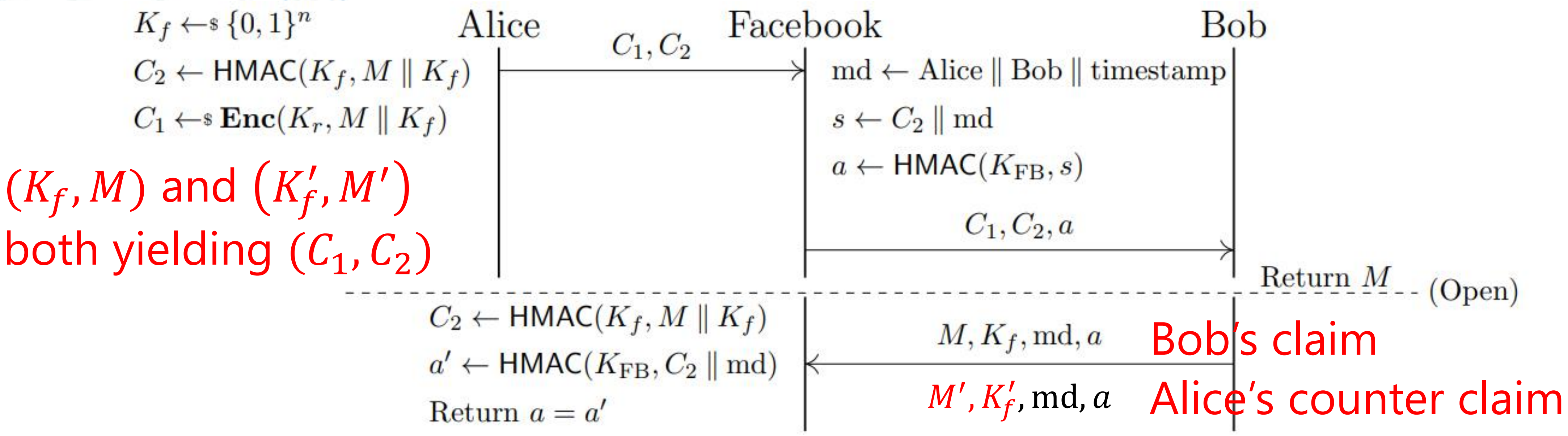


Figure 3: Facebook's message franking protocol [51]. The key K_r is a one-time-use symmetric key

(Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. “Message Franking via Committing Authenticated Encryption”)

- Bob reports to Facebook that Alice sent malicious M with K_F.
- Facebook checks the authenticity of Bob’s report, which is verified.
- Alice maliciously explains to Facebook that it was K_f' and M’.

Exploiting Lack of CMT Security by Alice (3/3)

NTT

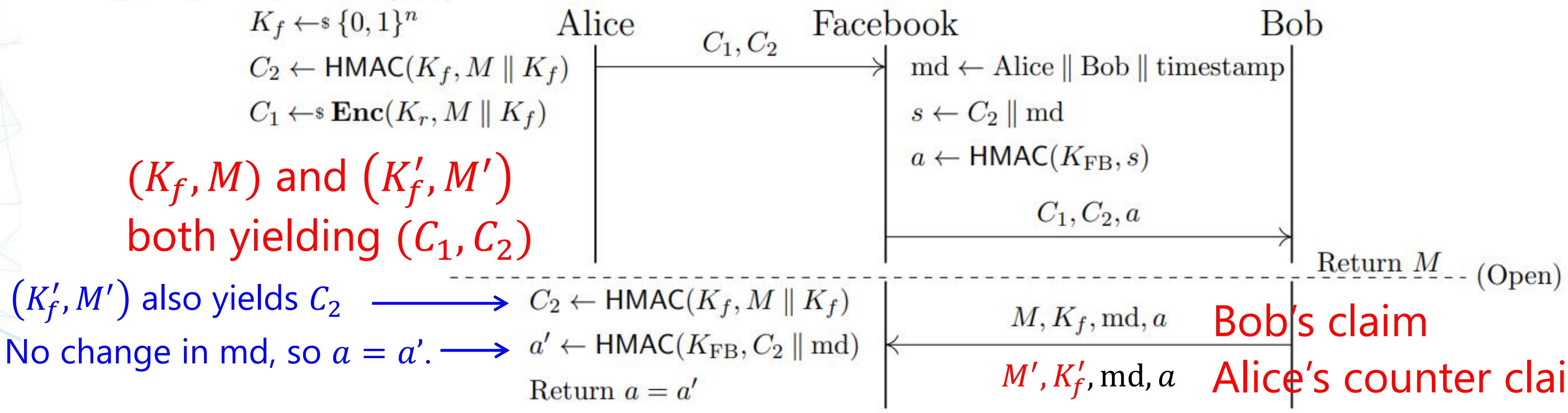


Figure 3: Facebook's message franking protocol [51]. The key K_r is a one-time-use symmetric key

(Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. "Message Franking via Committing Authenticated Encryption")

- Facebook checks the authenticity of Alice's report, which is also verified.
- Without CMT-security, malicious message report scheme doesn't work.

Key Commitment (CMT-1)

NTT

- Facebook's attempt is to verify the authenticity of not only the message but also the key by checking integrity.
- This is not a goal of integrity (abuse of symmetric-key crypto), which ensures the authenticity of the message under a fixed unknown key.
- In the context of public-key cryptography, the security notion for this setting is called **key commitment**.

An attacker cannot find a ciphertext decrypted with multiple keys, i.e., $\Pi_{\mathrm{Enc}}(K, N, A, M) = \Pi_{\mathrm{Enc}}(K', N, A, M)$ with $K \neq K'$.

Generalization: Context Committing (CMT-4)

NTT

- Generalization of key commitment by [BH22]

- Key commitment: K is different while N, A is the same:

$$\Pi_{\mathrm{Enc}}(K, N, A, M) = \Pi_{\mathrm{Enc}}(K', N, A, M') \text{ with } K \neq K'$$

- The natural extension is that any of K, N, A, M can be different, which is called **context commitment**.

$$\Pi_{\mathrm{Enc}}(K, N, A, M) = \Pi_{\mathrm{Enc}}(K', N', A', M') \text{ with} (K, N, A, M) \neq (K', N', A', M')$$

- No real application is known, but the context commitment achieves more robust security than the key commitment.

Generic and Dedicated CMT Security of AEAD Modes

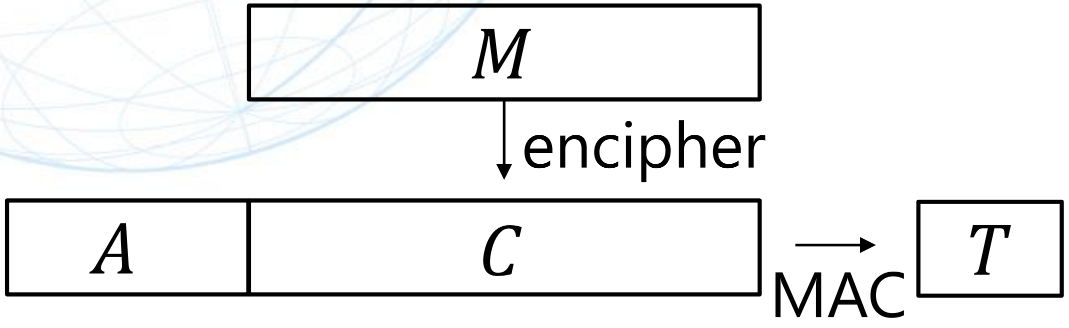
Desired Security Level for CMT-Security

- For CMT-security, the goal of the adversaries is to find key values generating a collision of the ciphertext.
- Everything can be computed offline.
- A typical attack scenario in the offline setting is the brute force attack on the key; k-bit security for k-bit key (128-bit security for AES-128 and 256-bit security for AES-256)
- Birthday-bound security of AES, 64 bits, is too small. At least 80-bit security is desired for CMT-security [CR22].

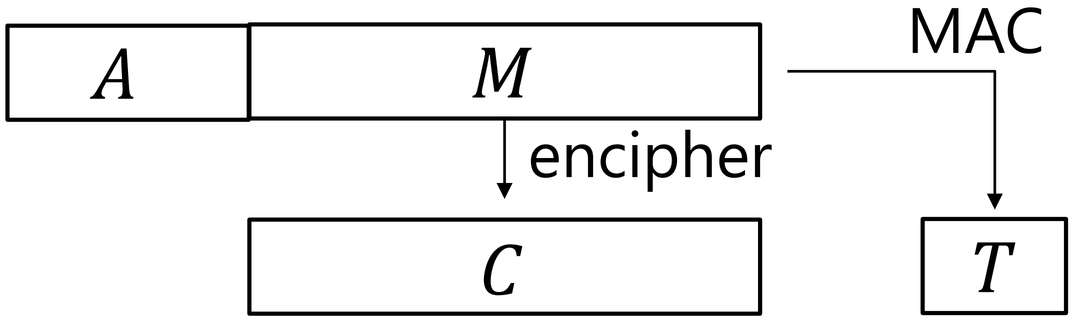
Generic Attack for Classes of AEAD

- Consider a class of AEAD s.t. A affects the tag generation but does not affect the message/plaintext conversion.

Enc-then-MAC e.g. GCM

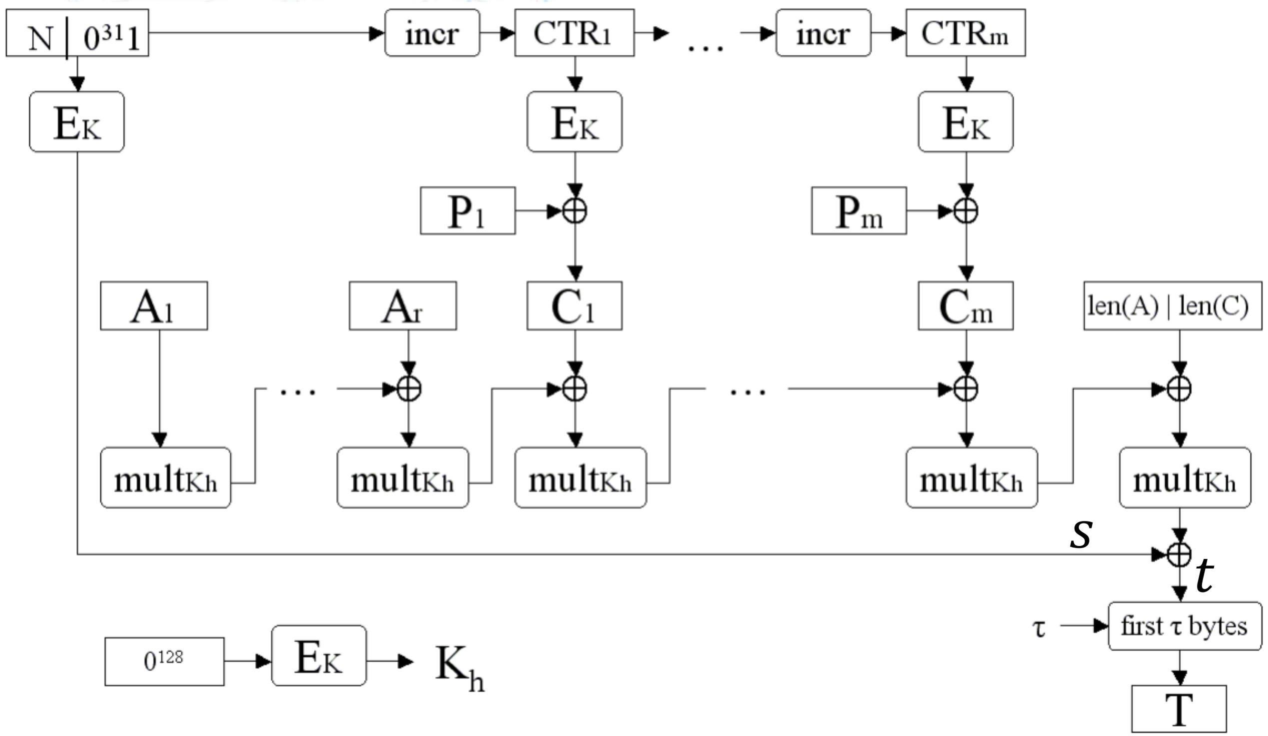


Enc-and-MAC e.g. OCB



- A generic attack with a cost of $2^{\frac{t}{2}}$, where t is a tag size, generates $\Pi_{\mathrm{Enc}}(K, N, A, M) = \Pi_{\mathrm{Enc}}(K', N, A', M')$.
- Find a tag collision between $Tag\ (K_1, N, A^i, C)$ and $Tag\ (K_2, N, A^j, C)$.

GCM is Not Key-Committing Secure

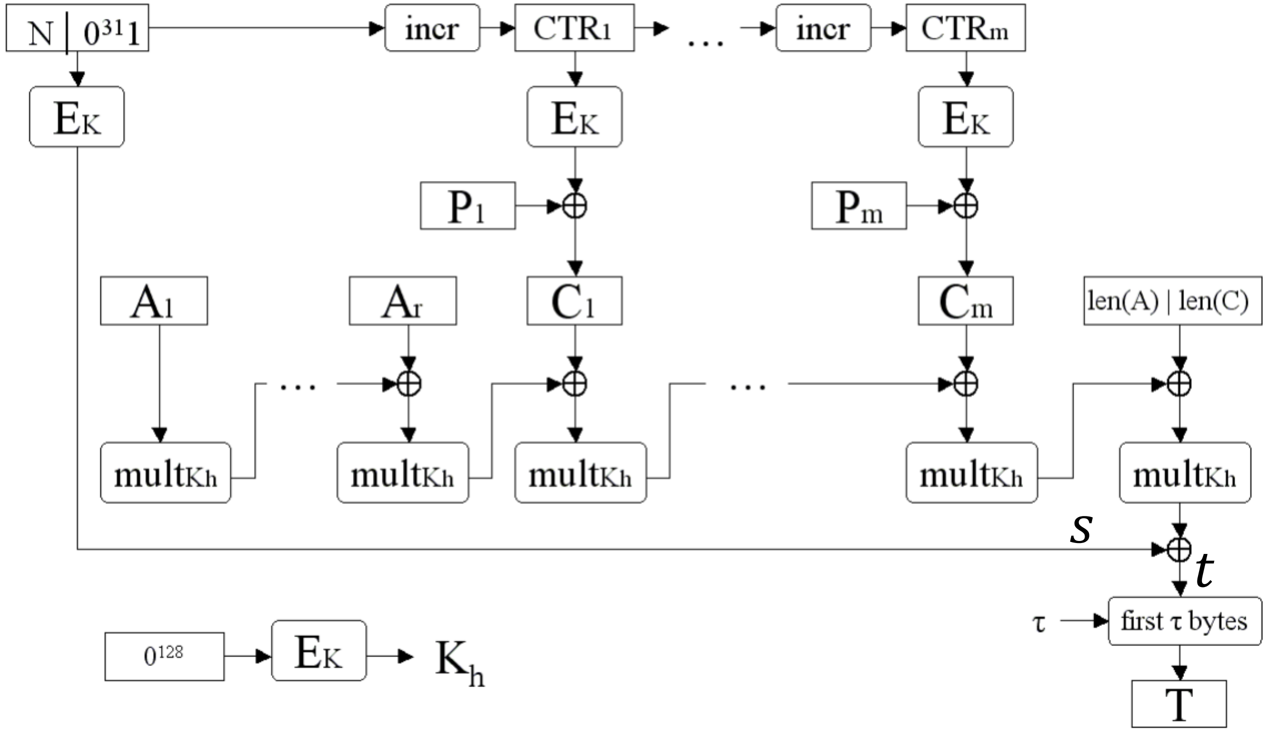


AES-GCM

- NIST SP800-38D
- Enc-then-MAC
  - Enc: AES-CTR
  - MAC: GMAC
- Allows the generic CMT-4 attack with 2^{64} cost.
- A constant time attack exists even for CMT-1.

Breaking Key-Committing Security of GCM

NTT



- Field multiplication in GHASH is invertible if a key is known.
- Easy to derive the same (C, T) for two keys.

Set $A \leftarrow \phi$. For given K_1, K_2, N, and C_i for all blocks but the j-th,

- Tag for K_1: $T_1 = s_1 + C_j \cdot K_{H_1}^{m-j} + \Sigma_{i=1,i\neq j}^{m} C_i \cdot K_{H_1}^{m-i}$
- Tag for K_2: $T_2 = s_2 + C_j \cdot K_{H_2}^{m-j} + \Sigma_{i=1,i\neq j}^{m} C_i \cdot K_{H_2}^{m-i}$

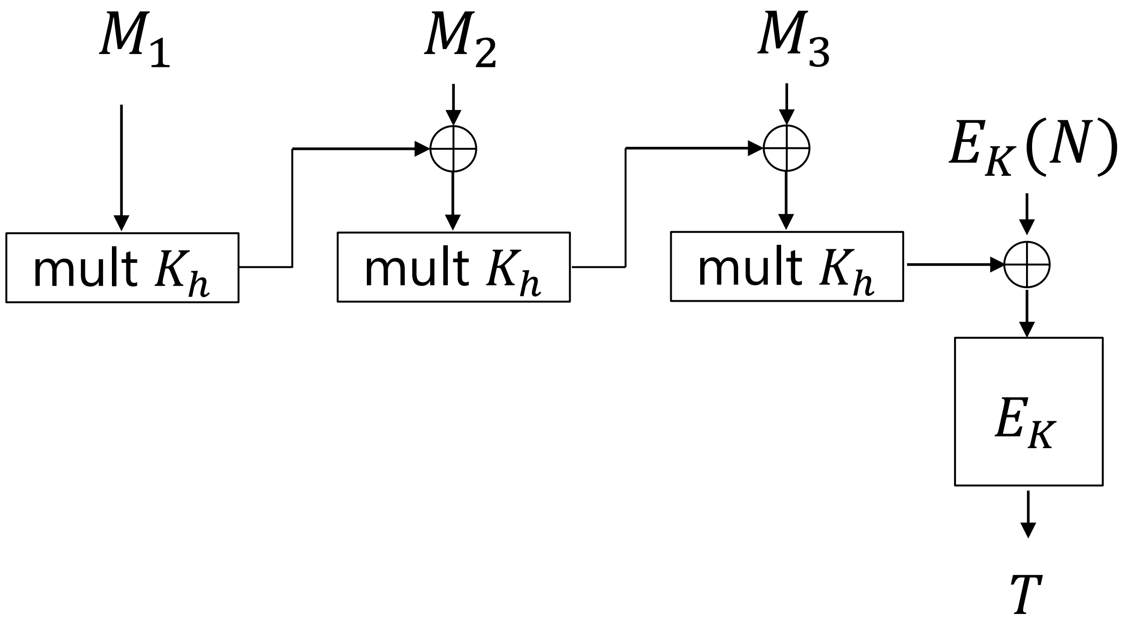
C_j is the only unknown variable. By setting $T_1 = T_2$, C_j is calculated.

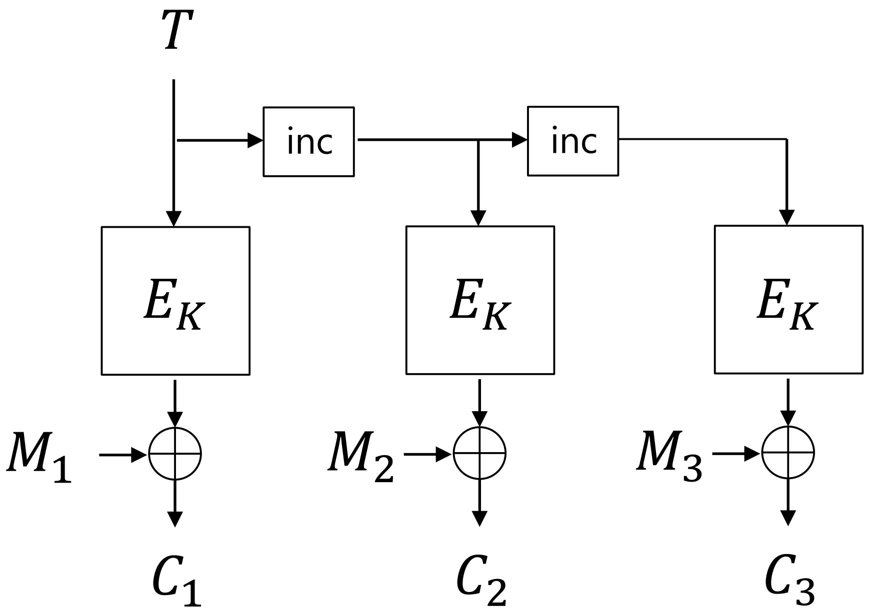
AES-GCM-SIV

NTT

- provide better nonce-misuse security than GCM
- standardized as RFC 8452
- SIV paradigm for MAC-then-ENC approach;

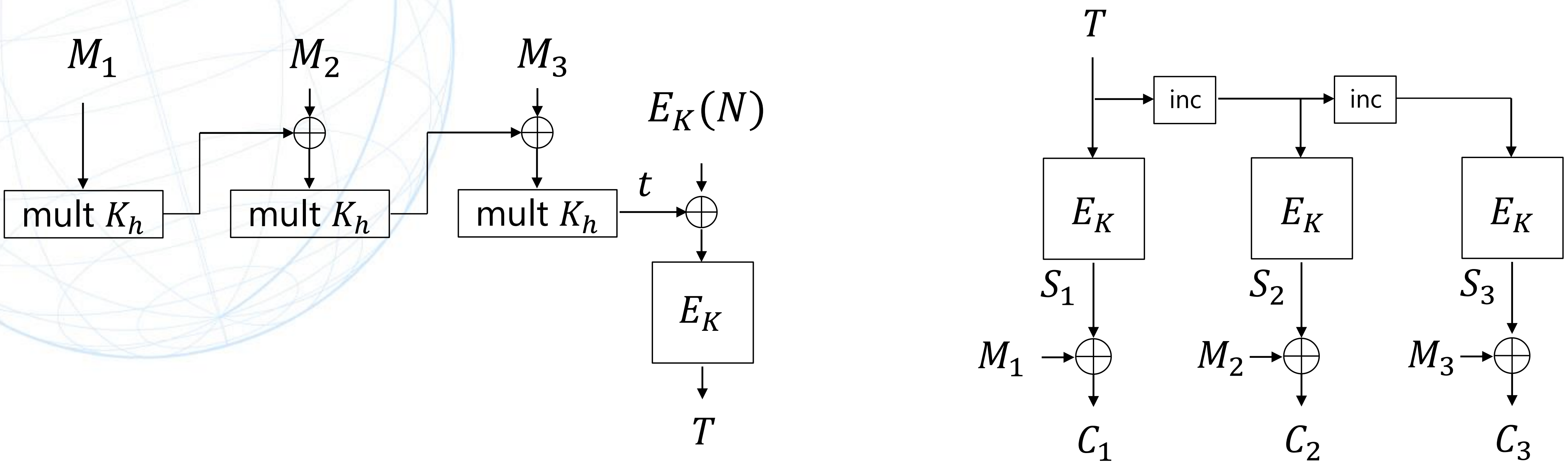
  the generic attack of cost $2^{t/2}$ is not applicable.
- A constant time attack exists even for CMT-1.





Breaking Key-Committing Sec of GCM-SIV

NTT



- Fix K, K', N, T, which fixes key streams S_i, S_i' and hash output t, t'.
- All constraints are linear: $t = \Sigma_{i=1}^{m} M_i \cdot K_h^{m-i}$, $t' = \Sigma_{i=1}^{m} M_i' \cdot K_h^{m-i}$,
  $M_i \oplus S_i = M_i' \oplus S_i'$ for $i = 1 \dots m$.
- $2m$ variables for $m + 2$ constraints. Easy to find the solution.

Summary of CMT-Security of AEAD Standard

- Basically, conventional AEAD schemes have not been designed by having CMT-security in mind.

- Most of the AEAD modes that current have been standardized can be attacked in some sense, which includes the following.
  - GCM
  - CCM
  - OCB
  - GCM-SIV
  - AEGIS

Robust AE and its CMT-Security

Practical Security Issues of Nonce-based AEAD

Nonce-based AEAD assumes that protocols provide a nonce that never takes the same value, achieving high security and high speed. However, nonce-based AEAD may be vulnerable for incorrect implementations.

Nonce misuse:

- Protocol designers may not be a crypto expert, and the same nonce may be repeated often. The worst case is that nonce is fixed to 0.

Decryption misuse:

- The decrypted M should be output only after the tag is verified. However, implementers may fail it, or storing huge amount of decryption results before the verification is impossible.

Robust AE

Robust AE resolves both issues of nonce- and decryption-misuses.

Encryption:

A single bit of change in any of N, A, M randomizes the whole C, T. The only information leak in nonce misuse is that exactly the same N, A, M is iteratively processed under the same K.

Decryption:

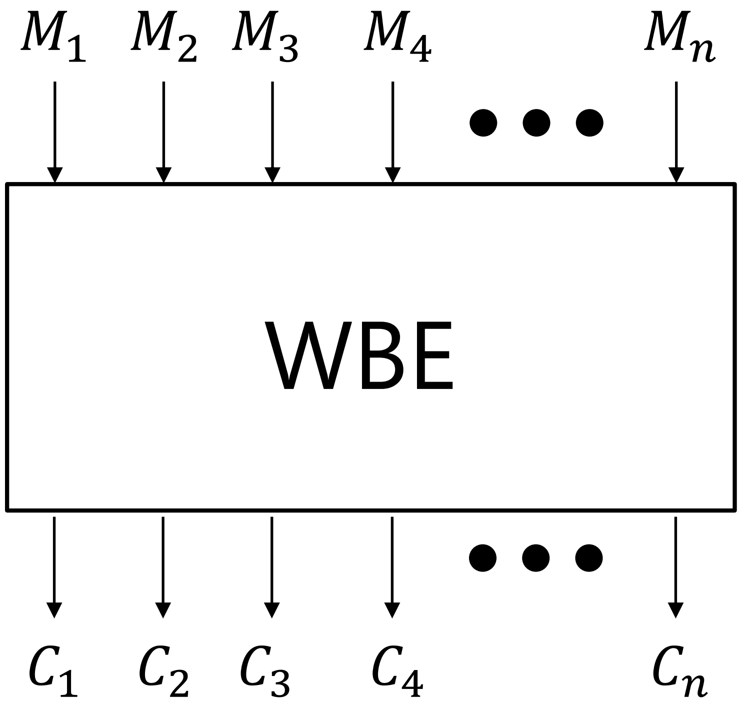
A single bit of change in any of N, A, C, T randomizes the whole M. Even the decrypted results are released without being verified, what the attacker receive is a random string.

Wide-Block Encryption and Robust AE

NTT

- Robust-AE can be constructed from wide-block encryption mode with encode-then-encipher paradigm.

- **Wide-block encryption:**
  - Entire construction behaves like a block cipher
  - Any change in M randomize the whole C.
  - Any change in C randomize the whole M.

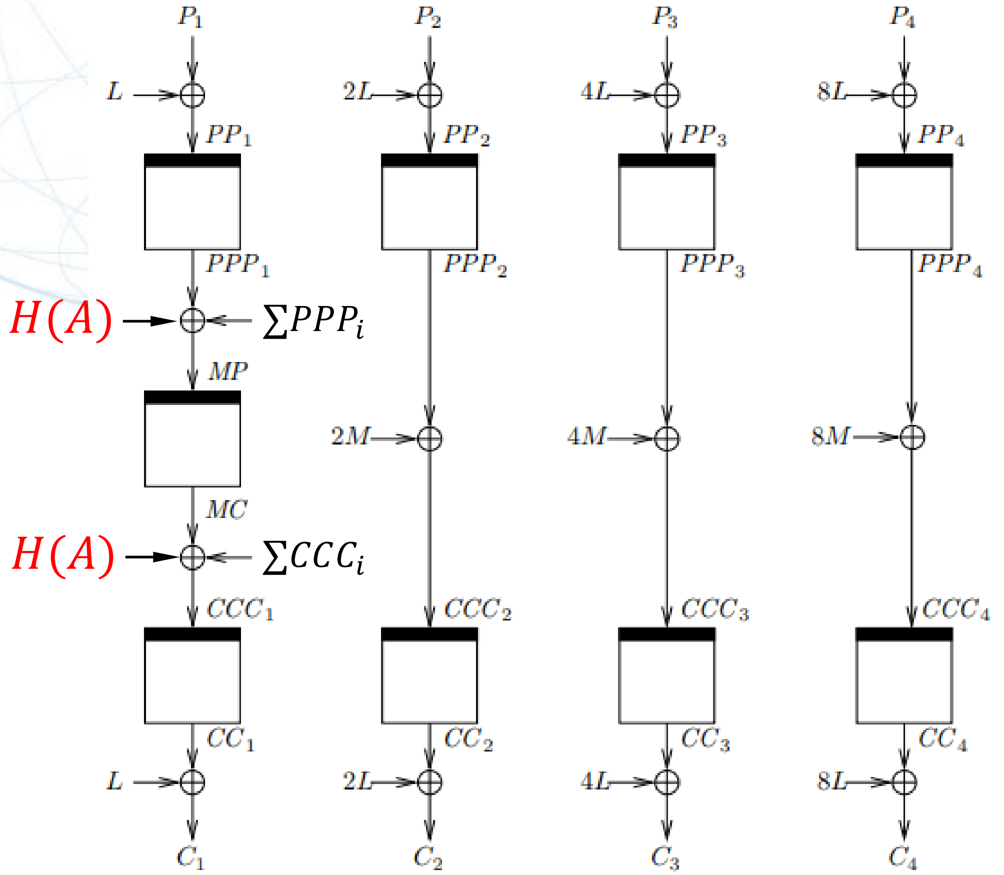


- **Encode-then-Encipher:**
  - Add zero bits to M.
  - Upon decryption, check if the added zeros are recovered.

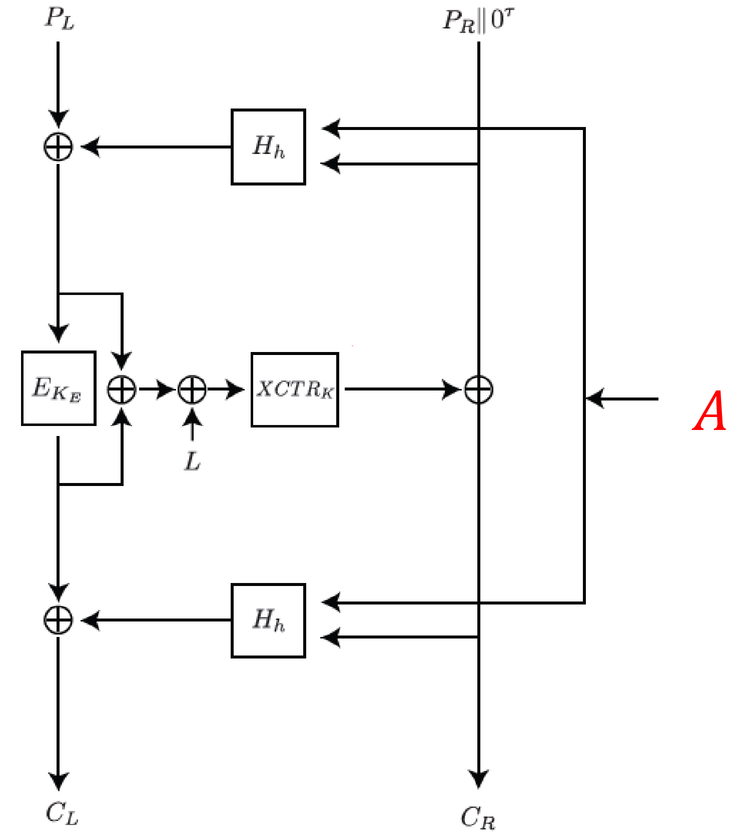
- NIST will standardize a WBE mode, accordion mode. CMT-security for AEADs built from the accordion mode is actively discussed.

Popular WBE: EME and HCTR2

- ECB-Mix-ECB (EME)
- A base of AEZ [HKR15]



- Hash-Encipher-Hash
- HCTR2 [CHB21] developed by Google is used in Android's file encryption.

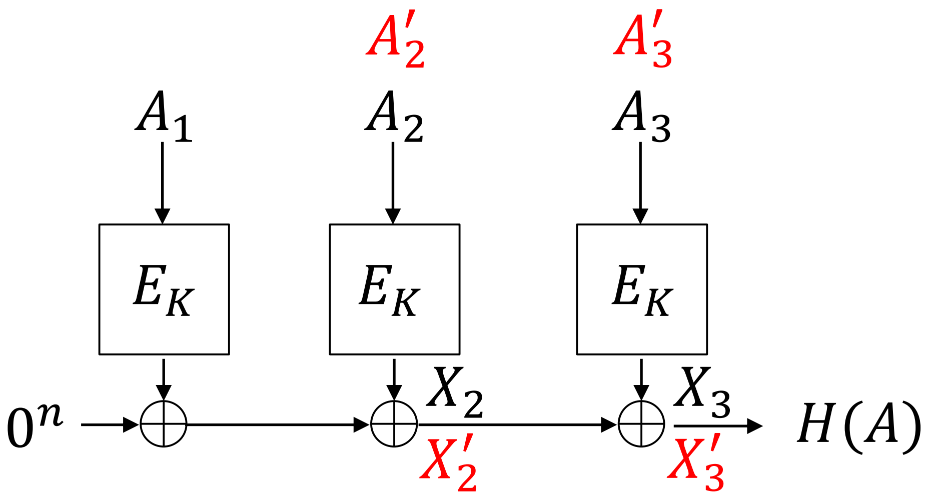


CMT-4 is broken if (A, A') s.t. $H(A) = H(A')$ is generated.

O(1) CMT-4 Attack for EME [CDD+24]

NTT

- CMT-4 is broken if (A, A') s.t. $H(A) = H(A')$ is generated.
- EME uses many E_K. Suppose that H is also based on E_k, particularly PHASH is used to parallel processing
- With the knowledge of K, E_K is invertible. Easy to modify the last two blocks of A to A' s.t. $H(A) = H(A')$.

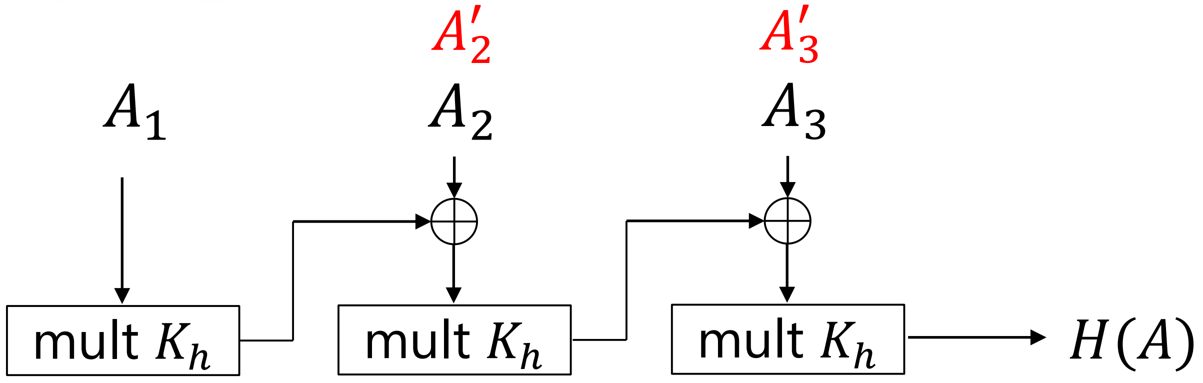


1. Modify A_2 to arbitrary A'_2.
2. Compute $X'_2 = E_K(A'_2)$.
3. We want X'_3 to be $X_3 \oplus X_2 \oplus X'_2$.
4. Compute $A'_3 = E_K^{-1}(X_3 \oplus X_2 \oplus X'_2)$

$$H(A_1||A_2||A_3) = H(A_1||A'_2||A'_3)$$

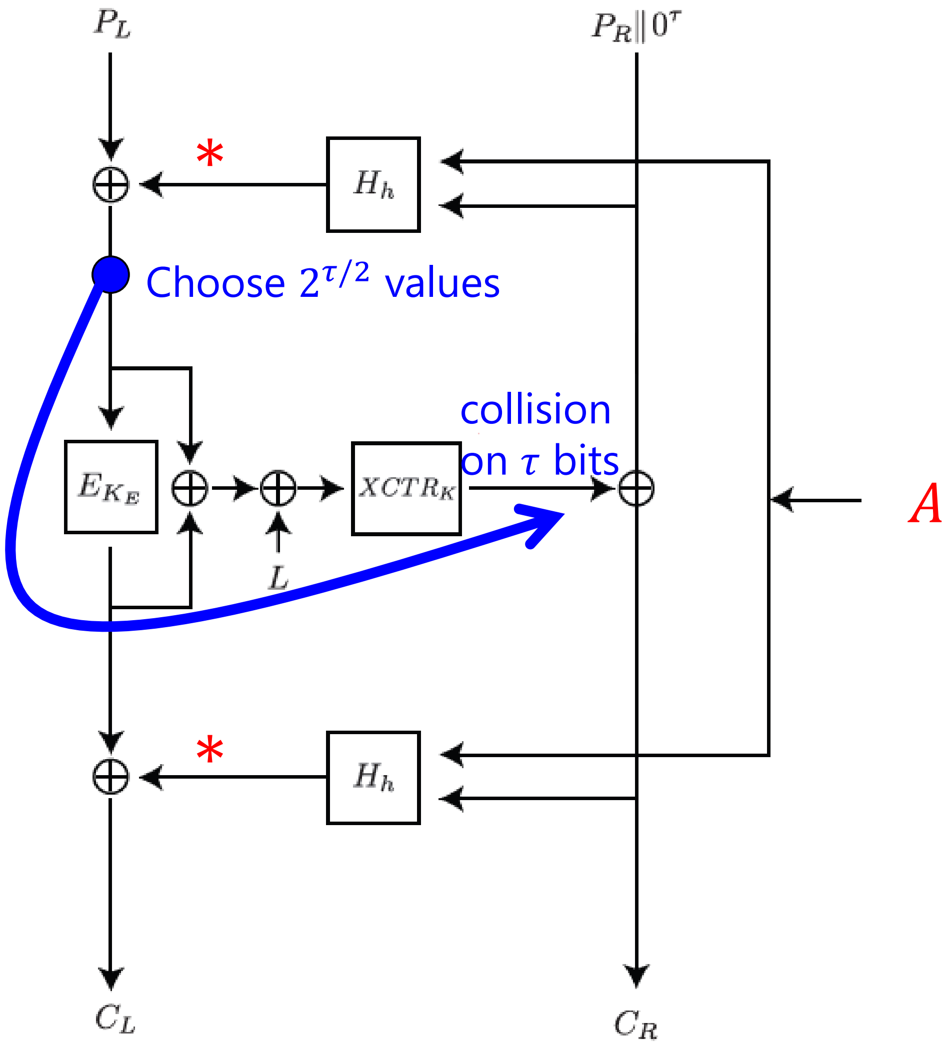
O(1) CMT-4 Attack for HCTR2-EtE [CDD+24]

- CMT-4 is broken if (A, A') s.t. $H(A) = H(A')$ is generated.
- The hash of HCTR2 is a polynomial hash, similar as GHASH.
- With the knowledge of K, it is easy to modify the last two blocks of A to A' s.t. $H(A) = H(A')$.



$O(2^{\tau/2})$ CMT-1 Attack for HCTR2-EtE [CDD+24]

- The last τ bits of plaintext is fixed to 0 for the encode-then-encipher.
- With the knowledge of the hash key, by choosing A, $H(A)$ can produce any output, namely $H(A)$ is invertible. Then, colliding C_L is always achieved by properly choosing P_L, P_L'.
- For the right branch, except for the last τ bits, colliding C_R can be achieved by properly choosing P_R, P_R'.
- Try $2^{\tau/2}$ values of the left-block to find a collision on the last τ bits of the XCTR.



On-going Recent Challenge

Committing Wide Encryption Mode with Minimum Ciphertext Expansion
Joint work with Yusuke Naito and Takeshi Sugawara
[ePrint 2024/1257]

Future Standardization of WBE by NIST

NIST will standardize a WBE scheme, accordion mode.

Research Challenge:

- How efficiently can we add CMT-4 security, the maximum CMT-security, by using a WBE as an underlying primitive?
- By appending $H(K, A)$ to a tag, CMT-4 security is added.

$$(K, A, M) \rightarrow (C, T, H(K, A))$$

  - Communication cost is heavier than computational cost. We aim the minimum ciphertext expansion.
  - In decryption, modifying $H(K, A)$ doesn't impact to M, thus it does not satisfy Robust-AE. The new construction should preserve the Robust-AE.

Comparison of Expansion Size and Security

NTT

- WBE+EtE (eg HCTR2+EtE) achieves birthday-bound CMT-1 security for the expansion size, and no CMT-4 security.
- Appending $H(K, A)$, CMT-4 is provided but the expansion is bigger.

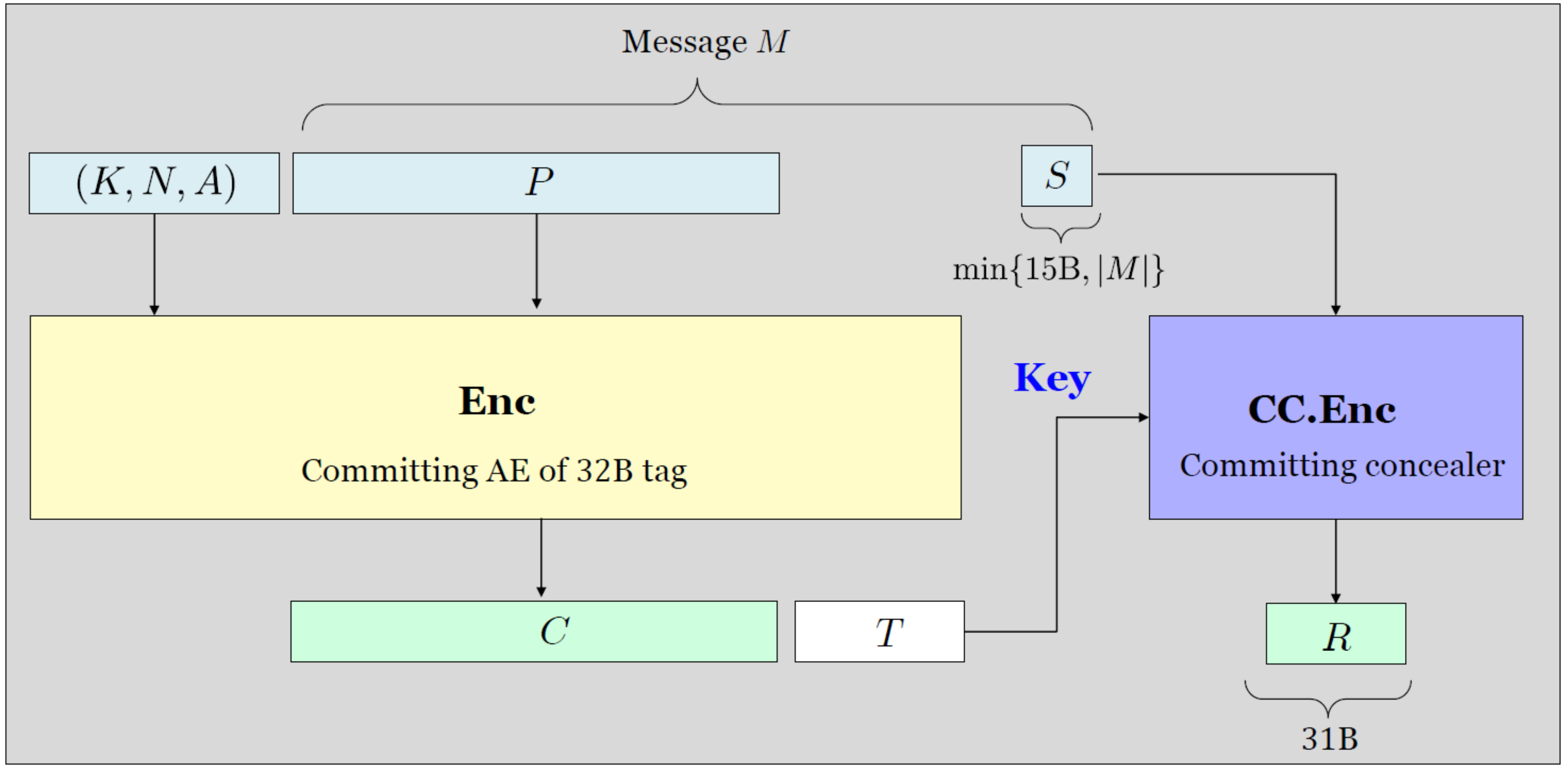
Scheme	Expansion bits	AE	CMT	Minimum?	Primitive	Ref.
WBE + EtE†	$2s_{\mathsf{cmt}}$‡	RAE	CMT-1	No	IC	[BR00, CFI+23]
Tag AE + CC	s_{cmt}	non-RAE	CMT-4	Yes	RO	[BHW23]
FFF	s_{cmt}	RAE	CMT-4	Yes	RO	Ours

† AEZ, Adiantum-EtE, HCTR2-EtE, ‡The block size of the internal block cipher is $2s_{\mathsf{cmt}}$ bits.

Approach for Minimal Ciphertext Expansion

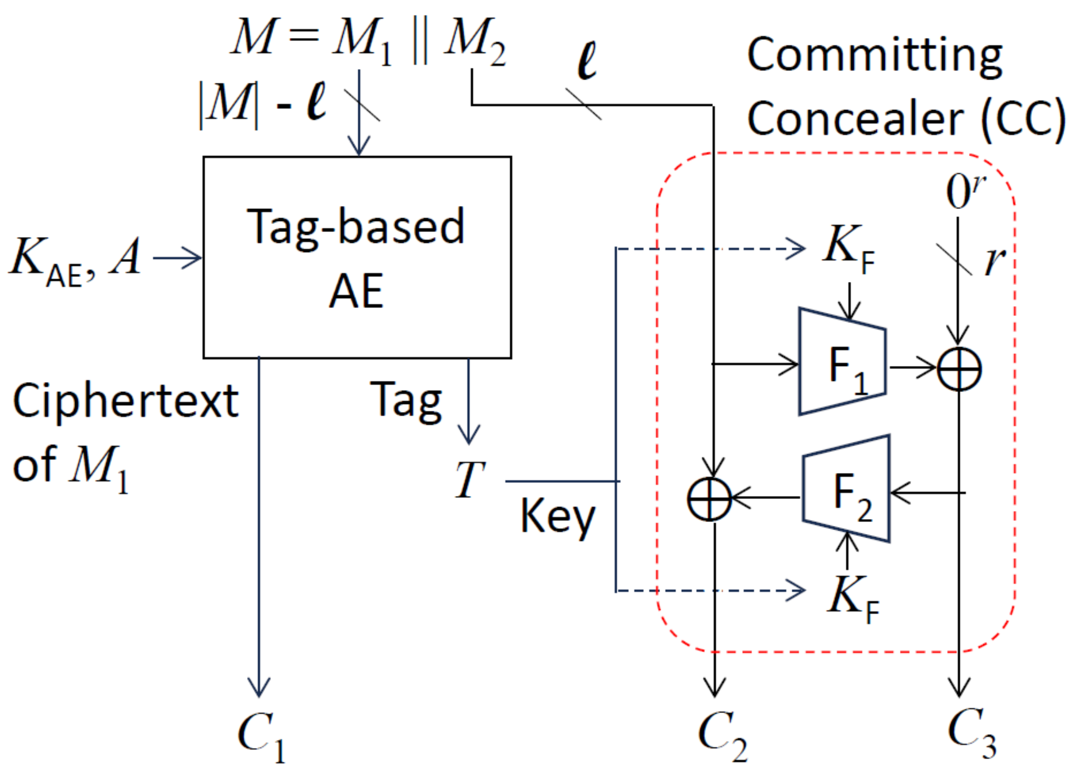
NTT

- Committing concealer [BTW23] at NIST workshop 2023
- Having 2ℓ bits of string is necessary to ensure ℓ-bit CMT-4 security.
- Having ℓ bits of redundancy is necessary to ensure ℓ-bit RAE security.
- Divide M to ℓ bits and the rest, recover ℓ bits of M during verification.



Our Instance of Committing Concealer

- For a tag-based AEAD, 2-round Feistel, known to achieve the birthday-bound collision resistance, is sufficient.
- Its direct application to WBE is not RAE.
- WBE has no tag, and an attempt to use a fraction of WE's ciphertext as a CC's key does not work because Dec first needs to recover CC's key from C_1.
- An adversary can distinguish the released unverified plaintexts because
  - (i) Dec of M_1 is unaffected by C_2, C_3
  - (ii) $\Delta C_2 = \Delta M_2$ with probability 1.



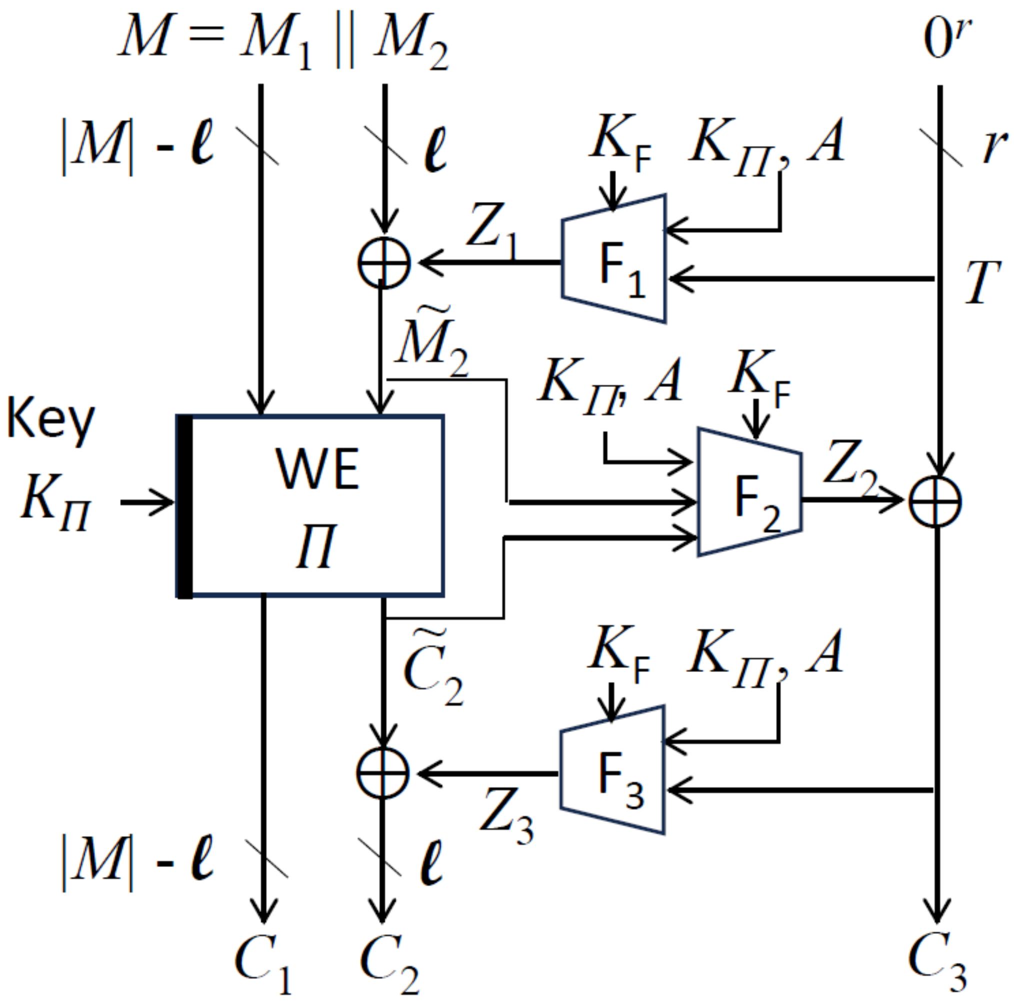
Our Construction

NTT

- Partitioning of M is similar to CC.

- Interaction between WE and CC is carefully designed, e.g.
  - without the line from $\widetilde{C_2}$ to F_2, encryption of M_1 does not impact C_3.
  - without the line from $\widetilde{M_2}$ to F_2, decryption of C_1 does affect verification.

- Achieve s_{cmt} CMT-4 security and s_{rae} RAE security with the ciphertext size only max$\{s_{cmt}, s_{rae}\}$ bits larger than the message size.



Conclusion

Conclusion

NTT

- Lack of CMT-1 security (key commitment) causes serious impact in some real-world use cases. CMT-4 security (context commitment) has not find real-world applications yet, but it is the highest security achieved.
- Conventional AEADs were not designed to provide CMT-security, and most of currently standardized AEADs can be broken terribly particularly CMT-4 security.
- WBE, or accordion mode, is a recent trend to be more robust in AEAD. WBE + CMT-security is an interesting research direction.

Thank you for your attention !!

References

[FOR17] Pooya Farshim, Claudio Orlandi, and Razvan Rosie. Security of symmetric primitives under incorrect usage of keys. IACR Trans. Symmetric Cryptol., 2017(1):449–473, 2017.

[GLR17] Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. Message franking via committing authenticated encryption. In CRYPTO 2017, pages 66–97, 2017.

[BH22] Mihir Bellare and Viet Tung Hoang. Efficient schemes for committing authenticated encryption. In EUROCRYPT 2022, volume 13276, pages 845–875, 2022.

[CR22] John Chan and Phillip Rogaway. On committing authenticated encryption. In ESORICS 2022, volume 13555, pages 275–294, 2022.

[HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated encryption AEZ and the problem that it solves. In EUROCRYPT 2015, volume 9056 of LNCS, pages 15–44, 2015.

[CHB21] Paul Crowley, Nathan Huckleberry, and Eric Biggers. Length-preserving encryption with HCTR2. IACR Cryptol. ePrint Arch., 2021.

[CDD+24] Yu Long Chen, Michael Davidson, Morris Dworkin, Jinkeon Kang, John Kelsey, Yu Sasaki, Meltem Sönmez Turan, Donghoon Chang, Nicky Mouha, and Alyssa Thompson. Proposal of requirements for an accordion mode. 2024.

[BHW23] Mihir Bellare, Viet Tung Hoang, and Cong Wu. The landscape of committing authenticated encryption (presentation at NIST Workshop 2023). 2023.