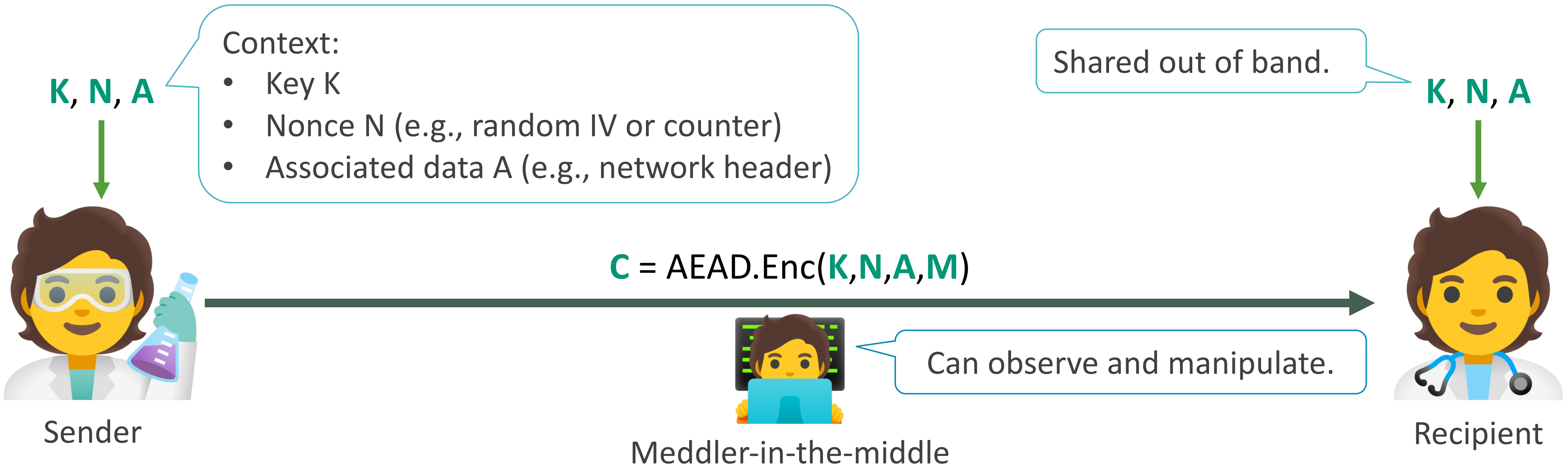
Context Discovery and Commitment Attacks

How to Break CCM, EAX, SIV, and More

Sanketh Menda, **Julia Len**, Paul Grubbs, and Thomas Ristenpart

Eurocrypt 2023

Authenticated Encryption with Associated Data (AEAD)



Standardized
1. AES-GCM
2. ChaCha20/Poly1305
3. AES-GCM-SIV

Provably Secure
1. Confidentiality
2. Authenticity

Recent Attacks on AEAD

These attacks work in new threat models!

These attacks exploit lack of *key commitment*:
An adversary can find keys K_1, K_2 and ciphertext C s.t. C can be decrypted under both keys

Context Commitment Security [BH22]

For AEAD, computationally efficient to find

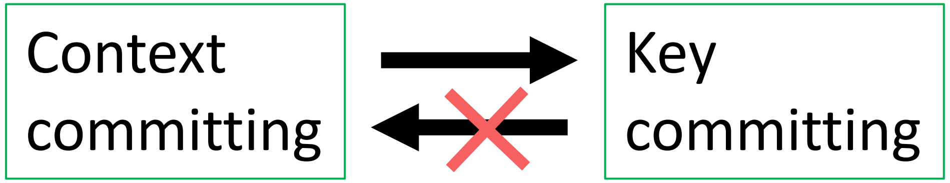
$(K_1, N_1, A_1), (K_2, N_2, A_2)$ and C

such that decryption

$M_1 \leftarrow \text{AEAD.Dec}(K_1, N_1, A_1, C)$

$M_2 \leftarrow \text{AEAD.Dec}(K_2, N_2, A_2, C)$

succeeds.



Commitment Attacks (Before this Paper)

Scheme	Key Committing	Context Committing
GCM	❌ [GLR17, DGRW19]	❌ ↪️
AES-SIV	❔	❔
CCM	❔	❔
EAX	❔	❔
OCB3	❌ [ADGKLS20]	❌ ↪️
PaddingZeros	✔ [ADGKLS20]	❔
KeyHashing	✔ [ADGKLS20]	❔
CAU-C1	✔ [BH22]	❔

Do CCM and EAX provide key commitment?

Asked 2 years, 2 months ago   Modified 2 years, 2 months ago   Viewed 171 times

10

In an interesting paper called "Partitioning Oracle Attacks" by Julia Len, Paul Grubbs & Thomas Ristenpart an attack is presented on 1.5 pass AEAD schemes that utilize GMAC (GCM, AES-GCM, AES-GCM-SIV) and Poly1305 which is often used with a ChaCha/Salsa variant.

In the paper they mention that older schemes based on HMAC authentication are not vulnerable against this attack because they provide the key commitment property.

Do CCM with CBC-MAC and EAX with AES-CMAC provide key commitment as well? Or is - for instance - the output size of the MAC constructions too small? If they don't provide full key commitment, are they susceptible to this attack?

cryptanalysis cbc-mac cmac decryption-oracle eax

Share Improve this question Follow

asked Jan 25, 2021 at 15:57
Maarten Bodewes ♦
89.5k 13 153 307

Sophie, indistinguishable from random noise @SchmiegSophie · Sep 7, 2020
Given that I've now done the work of setting up the blog, I used the chance to write a bit about invisible salamanders, i.e. ciphertexts that decrypt to two different plaintexts depending on the used key, for AES-GCM and AES-GCM-SIV.

keymaterial.net
Invisible Salamanders in AES-GCM-SIV
By now, many people have run across the Invisible Salamander paper about the interesting property of ...

4 40 128

JP Aumasson @veorq · Sep 9, 2020
Replying to @SchmiegSophie
no such trick for SIV-AES I guess?

1 2

Sophie, indistinguishable from random noise
@SchmiegSophie

Replying to @veorq

Yeah, for AES-SIV it's much harder. You can construct a collision for the CMAC, but it's non-linear, so the encrypted text would not collide easily.

Similar issue if you replace GMAC with Poly1305 in AES-GCM-SIV, because you mix characteristic.

1:44 PM · Sep 9, 2020 · Twitter for Android

Commitment Attacks (After this Paper)

Scheme	Key Committing	Context Committing
GCM	❌ [GLR17, DGRW19]	❌ ↪️
AES-SIV	❌ ✨	❌ ↪️ ✨
CCM	❌ ✨	❌ ↪️ ✨
EAX	❌ ✨	❌ ↪️ ✨
OCB3	❌ [ADGKLS20]	❌ ↪️
PaddingZeros	✔️ [ADGKLS20]	❌ ✨
KeyHashing	✔️ [ADGKLS20]	❌ ✨
CAU-C1	✔️ [BH22]	❌ ✨

✨ = **new result**

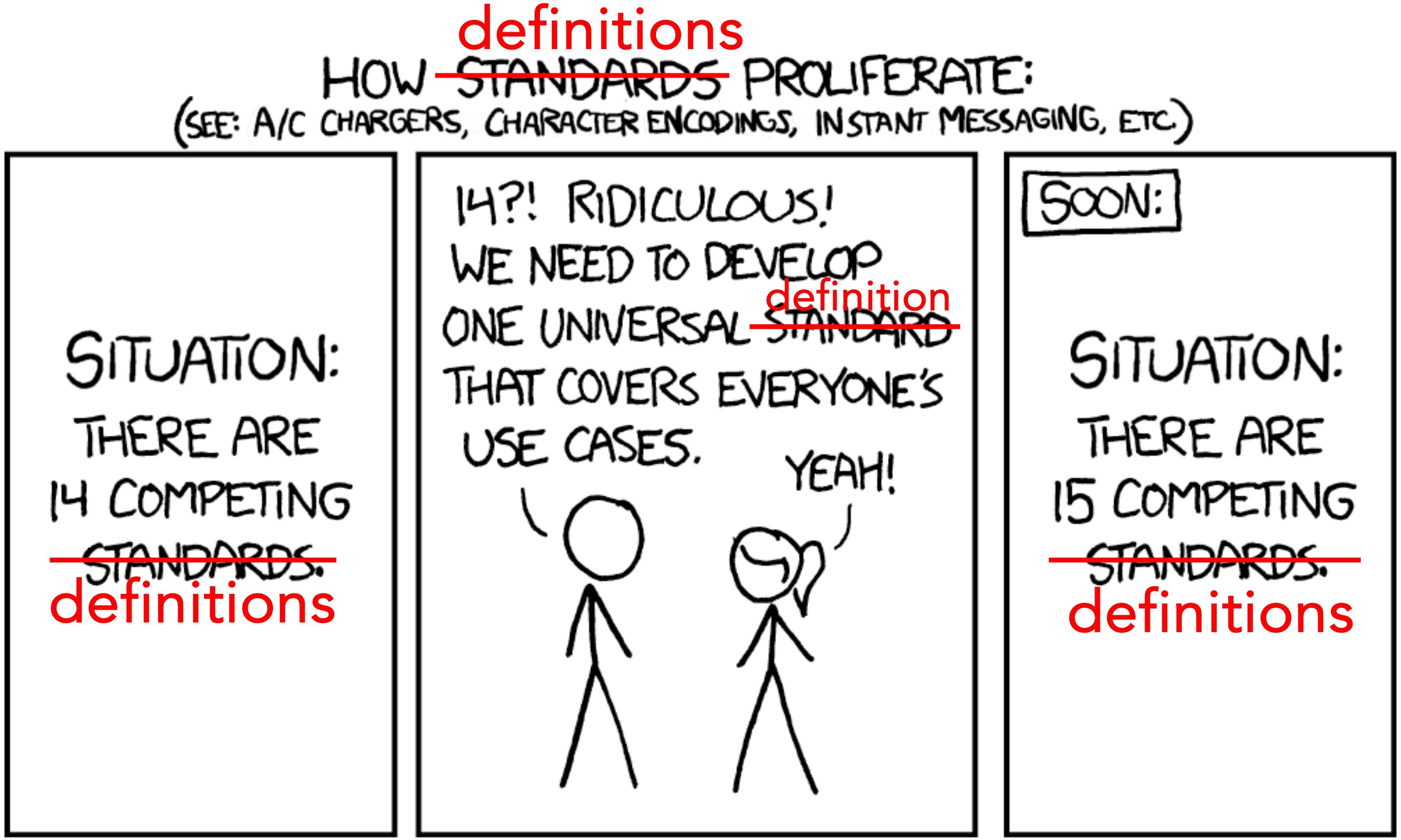
Our Contributions

- New granular framework for context commitment
- Key commitment attack against the original SIV mode
- New context commitment security notion: context discovery
- Context discovery attacks against GCM, OCB3, EAX, CCM, SIV

Our Contributions

- New granular framework for context commitment
- Key commitment attack against the original SIV mode
- New context commitment security notion: context discovery
- Context discovery attacks against GCM, OCB3, EAX, CCM, SIV

Don't we already have committing security definitions?



https://xkcd.com/927/

Ambiguity in Key Commitment Definitions

$\mathrm{FROB}^{\mathcal{A}}_{\mathbf{AE}}(\lambda)$:

$(C, K_1, K_2) \twoheadleftarrow \mathcal{A}(1^\lambda)$
if $K_1 = K_2$ return 0
$M_1 \leftarrow \mathbf{Dec}(K_1, C)$
$M_2 \leftarrow \mathbf{Dec}(K_2, C)$
return $(M_1 \neq \bot \wedge M_2 \neq \bot)$

[FOR17] doesn't mention associated data, and implicitly requires the same nonce.

$\mathrm{FROB}^{\mathcal{A}}_{\mathsf{SE}}$:

$((H, K), (H', K'), C) \leftarrow_{\$} \mathcal{A}$
If $K = K'$ then Return false
$M \leftarrow \mathsf{Dec}(K, H, C)$
$M' \leftarrow \mathsf{Dec}(K', H', C)$
Return $(M \neq \bot) \wedge (M' \neq \bot)$

[GLR17] allows different associated data, but still implicitly requires the same nonce.

We define targeted multi-key collision resistance (TMKCR) security by the following game. It is parameterized by a scheme AEAD and a target key set $\mathbb{K} \subseteq \mathcal{K}$. A possibly randomized adversary $\mathcal{A}$ is given input a target set $\mathbb{K}$ and must produce nonce N^*, associated data AD^*, and ciphertext C^* such that $\mathsf{AuthDec}_K(N^*, AD^*, C^*) \neq \bot$ for all $K \in \mathbb{K}$. We define the advantage via

$$\mathbf{Adv}^{\text{tmk-cr}}_{\mathsf{AEAD},\mathbb{K}}(\mathcal{A}) = \Pr\left[\mathrm{TMKCR}^{\mathcal{A}}_{\mathsf{AEAD},\mathbb{K}} \Rightarrow \mathsf{true}\right]$$

[LGR20] requires same nonces and associated data.

Granular Framework for Context Committing Security

For AEAD, computationally efficient to find

$(K_1, N_1, A_1), (K_2, N_2, A_2)$ and C

such that decryption

$M_1 \leftarrow \text{AEAD.Dec}(K_1, N_1, A_1, C)$

$M_2 \leftarrow \text{AEAD.Dec}(K_2, N_2, A_2, C)$

succeeds.

Granular Framework for Context Committing Security

Step 1: Arbitrary Predicates

For AEAD, computationally efficient to find

$(K_1, N_1, A_1), (K_2, N_2, A_2)$ and C

such that

$\mathbf{P}((K_1, N_1, A_1), (K_2, N_2, A_2))$ for predicate $\mathbf{P}$

and decryption

$\mathbf{M_1} \leftarrow \text{AEAD.Dec}(K_1, N_1, A_1, C)$

$\mathbf{M_2} \leftarrow \text{AEAD.Dec}(K_2, N_2, A_2, C)$

succeeds.

Granular Framework for Context Committing Security

Step 1: Arbitrary Predicates

For AEAD, computationally efficient to find

$(K_1, N_1, A_1), (K_2, N_2, A_2)$ and C

such that

$\mathbf{P}((K_1, N_1, A_1), (K_2, N_2, A_2))$ for predicate **P**

and decryption

$\mathbf{M_1} \leftarrow \text{AEAD.Dec}(K_1, N_1, A_1, C)$

$\mathbf{M_2} \leftarrow \text{AEAD.Dec}(K_2, N_2, A_2, C)$

succeeds.

Notion	Predicate
Context Commitment	$(K_1, N_1, A_1) \neq (K_2, N_2, A_2)$

Granular Framework for Context Committing Security

Step 1: Arbitrary Predicates

For AEAD, computationally efficient to find

$(K_1, N_1, A_1), (K_2, N_2, A_2)$ and C

such that

$\mathbf{P}((K_1, N_1, A_1), (K_2, N_2, A_2))$ for predicate $\mathbf{P}$

and decryption

$\mathbf{M_1} \leftarrow \text{AEAD.Dec}(K_1, N_1, A_1, C)$

$\mathbf{M_2} \leftarrow \text{AEAD.Dec}(K_2, N_2, A_2, C)$

succeeds.

	Notion	Predicate
	Context Commitment	$(K_1, N_1, A_1) \neq (K_2, N_2, A_2)$
Permissive	CMT-k	$K_1 \neq K_2$
Permissive	CMT-n	$N_1 \neq N_2$
Permissive	CMT-a	$A_1 \neq A_2$

Granular Framework for Context Committing Security

Step 1: Arbitrary Predicates

For AEAD, computationally efficient to find

$(K_1, N_1, A_1), (K_2, N_2, A_2)$ and C

such that

$\mathbf{P}((K_1, N_1, A_1), (K_2, N_2, A_2))$ for predicate $\mathbf{P}$

and decryption

$M_1 \leftarrow \text{AEAD.Dec}(K_1, N_1, A_1, C)$

$M_2 \leftarrow \text{AEAD.Dec}(K_2, N_2, A_2, C)$

succeeds.

	Notion	Predicate
	Context Commitment	$(K_1, N_1, A_1) \neq (K_2, N_2, A_2)$
Permissive	CMT-k	$K_1 \neq K_2$
	CMT-n	$N_1 \neq N_2$
	CMT-a	$A_1 \neq A_2$
Restrictive	CMT-k*	$K_1 \neq K_2 \wedge (N_1, A_1) = (N_2, A_2)$
	CMT-n*	$N_1 \neq N_2 \wedge (K_1, A_1) = (K_2, A_2)$
	CMT-a*	$A_1 \neq A_2 \wedge (K_1, N_1) = (K_2, N_2)$

Granular Framework for Context Committing Security

Step 2: Target Selection

For AEAD, given

$K_1, K_2 \leftarrow\$ \{0,1\}^k$ [target selection]

computationally efficient to find

$(K_1, N_1, A_1), (K_2, N_2, A_2)$ and C

such that

$\mathbf{P}((K_1, N_1, A_1), (K_2, N_2, A_2))$ for predicate **P**

and decryption

$M_1 \leftarrow \text{AEAD.Dec}(K_1, N_1, A_1, C)$

$M_2 \leftarrow \text{AEAD.Dec}(K_2, N_2, A_2, C)$

succeeds.

	Notion	Predicate
	Context Commitment	$(K_1, N_1, A_1) \neq (K_2, N_2, A_2)$
Permissive	CMT-k	$K_1 \neq K_2$
Permissive	CMT-n	$N_1 \neq N_2$
Permissive	CMT-a	$A_1 \neq A_2$
Restrictive	CMT-k*	$K_1 \neq K_2 \wedge (N_1, A_1) = (N_2, A_2)$
Restrictive	CMT-n*	$N_1 \neq N_2 \wedge (K_1, A_1) = (K_2, A_2)$
Restrictive	CMT-a*	$A_1 \neq A_2 \wedge (K_1, N_1) = (K_2, N_2)$

Granular Framework for Context Committing Security

Step 2: Target Selection

For AEAD, given

$C \leftarrow \text{AEAD.Enc}(K_1, N_1, A_1, M_1)$ [target selection]

and not given

K_1, N_1, A_1 [target hiding]

computationally efficient to find

(K_2, N_2, A_2)

such that

P$((K_1, N_1, A_1), (K_2, N_2, A_2))$ for predicate **P**

and decryption

$M_1 \leftarrow \text{AEAD.Dec}(K_1, N_1, A_1, C)$

$M_2 \leftarrow \text{AEAD.Dec}(K_2, N_2, A_2, C)$

succeeds.

	Notion	Predicate
	Context Commitment	$(K_1, N_1, A_1) \neq (K_2, N_2, A_2)$
Permissive	CMT-k	$K_1 \neq K_2$
Permissive	CMT-n	$N_1 \neq N_2$
Permissive	CMT-a	$A_1 \neq A_2$
Restrictive	CMT-k*	$K_1 \neq K_2 \land (N_1, A_1) = (N_2, A_2)$
Restrictive	CMT-n*	$N_1 \neq N_2 \land (K_1, A_1) = (K_2, A_2)$
Restrictive	CMT-a*	$A_1 \neq A_2 \land (K_1, N_1) = (K_2, N_2)$

Granular Framework for Context Committing Security

Step 2: Target Selection

For AEAD, given

$C \leftarrow \text{AEAD.Enc}(K_1, N_1, A_1, M_1)$ [target selection]

and not given

K_1, N_1, A_1 [target hiding]

computationally efficient to find

(K_2, N_2, A_2)

such that

$\mathbf{P}((K_1, N_1, A_1), (K_2, N_2, A_2))$ for predicate **P**

and decryption

$M_1 \leftarrow \text{AEAD.Dec}(K_1, N_1, A_1, C)$

$M_2 \leftarrow \text{AEAD.Dec}(K_2, N_2, A_2, C)$

succeeds.

	Notion	Predicate
	Context Commitment	$(K_1, N_1, A_1) \neq (K_2, N_2, A_2)$
Permissive	CMT-k	$K_1 \neq K_2$
Permissive	CMT-n	$N_1 \neq N_2$
Permissive	CMT-a	$A_1 \neq A_2$
Restrictive	CMT-k*	$K_1 \neq K_2 \land (N_1, A_1) = (N_2, A_2)$
Restrictive	CMT-n*	$N_1 \neq N_2 \land (K_1, A_1) = (K_2, A_2)$
Restrictive	CMT-a*	$A_1 \neq A_2 \land (K_1, N_1) = (K_2, N_2)$

This looks like preimage resistance.

Our Contributions

New granular framework for context commitment

Key commitment attack against the original SIV mode

New context commitment security notion: context discovery

Context discovery attacks against GCM, OCB3, EAX, CCM, SIV

Our Contributions

New granular framework for context commitment

Key commitment attack against the original SIV mode

New context commitment security notion: context discovery

Context discovery attacks against GCM, OCB3, EAX, CCM, SIV

SIV vs. GCM for CMT-k* attacks

Recall: CMT-k* means nonces and associated data of both contexts must be the same

GCM uses a highly-structured polynomial MAC

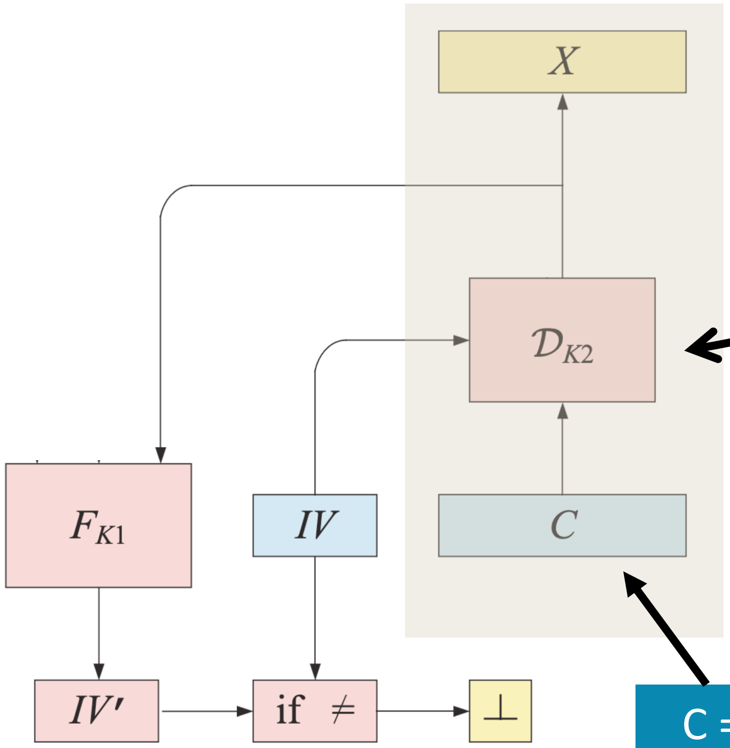
Finding CMT-k* attack is *easy*: this is just solving a simple system of 2 linear equations

SIV does not use a polynomial MAC so we can't adapt the attacks we already have…

The Original SIV Mode

For simplicity, we assume no associated data

CTR mode with “synthetic IV” in ciphertext as initial counter



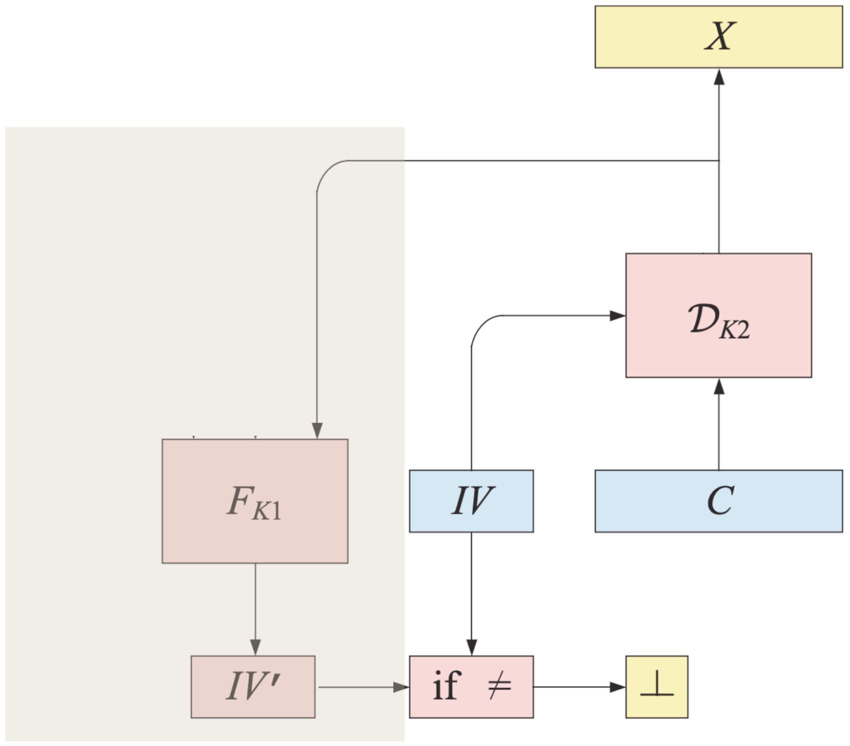
C = C’ || IV

1. Use CTR mode with key K2 to decrypt the ciphertext and recover message.
2. Recompute the “synthetic IV” from the message using S2V[CMAC] with key K1.
3. Compare this computed synthetic IV with that stored as part of the ciphertext:
   a) If they are different, then reject.
   b) Otherwise, return message.

[RS07] https://web.cs.ucdavis.edu/~rogaway/papers/siv.pdf

The Original SIV Mode

For simplicity, we assume no associated data

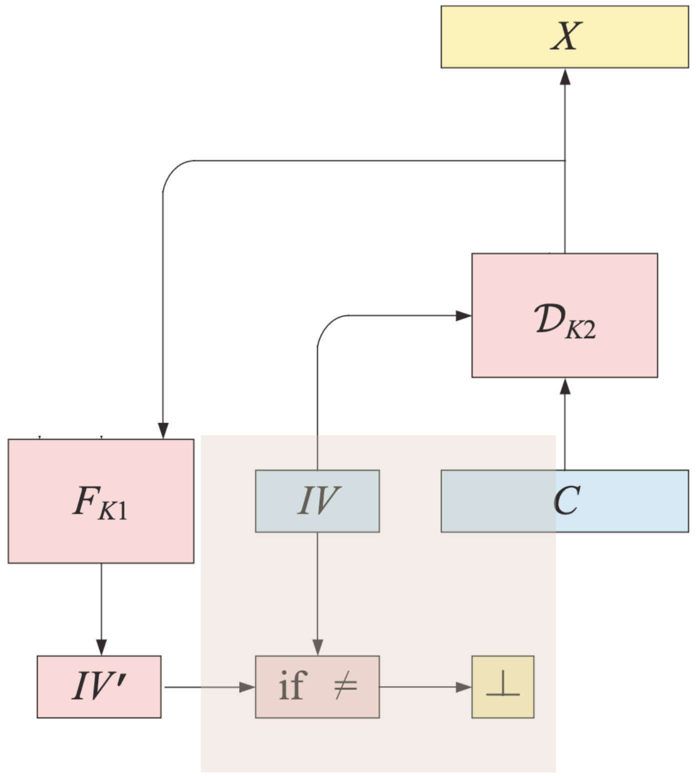


1. Use CTR mode with key K2 to decrypt the ciphertext and recover message.
2. Recompute the "synthetic IV" from the message using S2V[CMAC] with key K1.
3. Compare this computed synthetic IV with that stored as part of the ciphertext:
   a) If they are different, then reject.
   b) Otherwise, return message.

[RS07] https://web.cs.ucdavis.edu/~rogaway/papers/siv.pdf

The Original SIV Mode

For simplicity, we assume no associated data



1. Use CTR mode with key K2 to decrypt the ciphertext and recover message.
2. Recompute the “synthetic IV” from the message using S2V[CMAC] with key K1.
3. Compare this computed synthetic IV with that stored as part of the ciphertext:
   a) If they are different, then reject.
   b) Otherwise, return message.

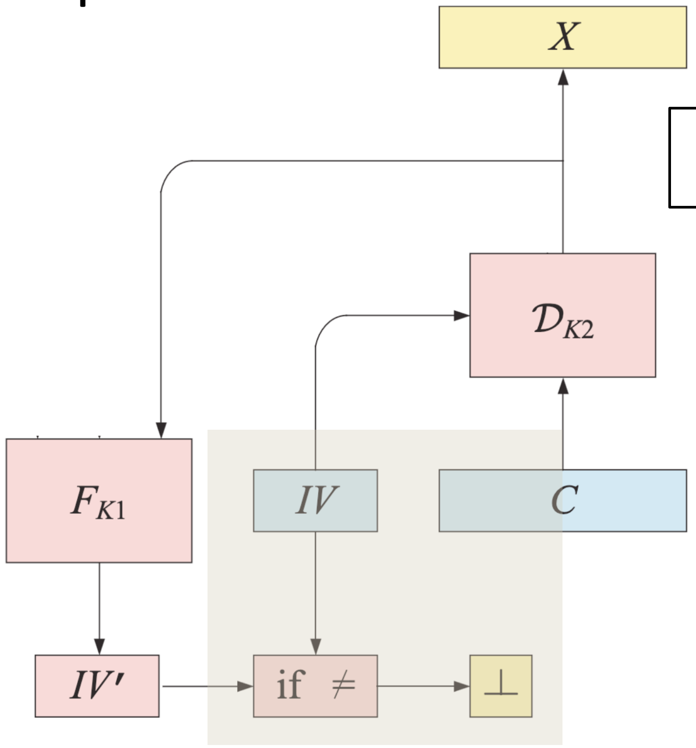
[RS07] https://web.cs.ucdavis.edu/~rogaway/papers/siv.pdf

CMT-k* attack on SIV

For simplicity, we'll consider 1-block ciphertexts with no associated data or nonce

Goal: Find two keys (K1, K2) ≠ (K1', K2') and ciphertext C (of the form C' || IV) so that decryption of C under both keys succeeds

- This means that the computed synthetic IV's match the stored IV that is part of ciphertext C



$$IV = CMAC\big(K1, C' \oplus E_{K2}(IV)\big) = CMAC\big(K1', C' \oplus E_{K2'}(IV)\big)$$

CMT-k* attack on SIV

For simplicity, we'll consider 1-block ciphertexts with no associated data or nonce

Goal: Find two keys (K1, K2) ≠ (K1', K2') and ciphertext C ←C' || IV such that:

$$\left(E_{K1}^{-1}(IV) \oplus \left(2 \cdot E_{K1}(0^n)\right) \oplus E_{K1}\left(2 \cdot E_{K1}(0^n)\right)\right) \oplus E_{K2}(IV)$$

$$\oplus \left(E_{K1'}^{-1}(IV) \oplus \left(2 \cdot E_{K1'}(0^n)\right) \oplus E_{K1'}\left(2 \cdot E_{K1'}(0^n)\right)\right) \oplus E_{K2'}(IV) = 0^n$$

If we model block cipher E as an ideal cipher, then this looks *very* close to the Generalized Birthday Problem!

CMT-k* attack on SIV: Using Generalized Birthday Problem

For simplicity, we'll consider 1-block ciphertexts with no associated data or nonce

4-list Birthday Problem:
Given lists L1, L2, L3, L4 of elements drawn uniformly and independently at random from $\{0,1\}^n$, find x1 ∈ L1, x2 ∈ L2, x3 ∈ L3, x4 ∈ L4 s.t. $\mathbf{x1 \oplus x2 \oplus x3 \oplus x4 = 0^n}$

Wagner gives the *k-tree algorithm* to solve this in $O(2^{n/3})$ space and time [W02]

$$\left(E_{K1}^{-1}(IV) \oplus (2 \cdot E_{K1}(0^n)) \oplus E_{K1}(2 \cdot E_{K1}(0^n))\right) \oplus E_{K2}(IV)$$
$$\oplus \left(E_{K1'}^{-1}(IV) \oplus (2 \cdot E_{K1'}(0^n)) \oplus E_{K1'}(2 \cdot E_{K1'}(0^n))\right) \oplus E_{K2'}(IV) = 0^n$$

$$F_1(K1) \oplus F_2(K2) \oplus F_3(K1') \oplus F_4(K2') = 0^n$$

<u>Problem</u>: These values are not drawn uniformly and independently at random from $\{0,1\}^n$

CMT-k* attack on SIV: Using Generalized Birthday Problem

For simplicity, we'll consider 1-block ciphertexts with no associated data or nonce

- We show that we can upper bound the distinguishability between the distribution formed by the values chosen to make the list and uniformly random distribution
- We then show we can apply Wagner's k-tree algorithm with the distributions we have and still have high probability of finding collisions
- We show that with high probability we can find a collision in time ~2^{53}, making it practical and sufficiently damaging to rule out SIV as suitable for contexts where key commitment matters

Our Contributions

New granular framework for context commitment

Key commitment attack against the original SIV mode

New context commitment security notion: context discovery

Context discovery attacks against GCM, OCB3, EAX, CCM, SIV

Our Contributions

New granular framework for context commitment

Key commitment attack against the original SIV mode

New context commitment security notion: context discovery

Context discovery attacks against GCM, OCB3, EAX, CCM, SIV

Revisiting the Framework for Context Committing Security

For AEAD, given

$C \leftarrow \text{AEAD.Enc}(K_1, N_1, A_1, M_1)$ [target selection]

and not given

K_1, N_1, A_1 [target hiding]

computationally efficient to find

(K_2, N_2, A_2)

such that

P$((K_1, N_1, A_1), (K_2, N_2, A_2))$ for predicate **P**

and decryption

$M_1 \leftarrow \text{AEAD.Dec}(K_1, N_1, A_1, C)$

$M_2 \leftarrow \text{AEAD.Dec}(K_2, N_2, A_2, C)$

succeeds.

	Notion	Predicate
	Context Commitment	$(K_1, N_1, A_1) \neq (K_2, N_2, A_2)$
Permissive	CMT-k	$K_1 \neq K_2$
	CMT-n	$N_1 \neq N_2$
	CMT-a	$A_1 \neq A_2$
Restrictive	CMT-k*	$K_1 \neq K_2 \wedge (N_1, A_1) = (N_2, A_2)$
	CMT-n*	$N_1 \neq N_2 \wedge (K_1, A_1) = (K_2, A_2)$
	CMT-a*	$A_1 \neq A_2 \wedge (K_1, N_1) = (K_2, N_2)$

This looks like preimage resistance.

Context Discoverability Security

CDY Security

For AEAD, given

C [target selection]

computationally efficient to find

K, **N**, **A**

such that decryption

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

succeeds.

Context Discoverability Security

CDY Security

For AEAD, given

C [target selection]

computationally efficient to find

K, **N**, **A**

such that decryption

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

succeeds.

- Context Discoverability security is to Context Committing security for AEAD as preimage resistance is to collision-resistance for hash functions
- We also show that if an AEAD scheme is “context compressing” (meaning: ciphertexts are decryptable under more than one context), then Context Committing security implies Context Discoverability security
- We show Context Discoverability attacks for CCM, EAX, SIV, GCM, and OCB3
- Context Discoverability allows us to better communicate attacks and threat models

Conclusion

Full version available on eprint:
https://ia.cr/2023/526

Scheme	Key Committing	Context Committing
GCM	[GLR17, DGRW19]	↪️
AES-SIV	✨	↪️ ✨
CCM	✨	↪️ ✨
EAX	✨	↪️ ✨
OCB3	[ADGKLS20]	↪️
PaddingZeros	[ADGKLS20]	✨
KeyHashing	[ADGKLS20]	✨
CAU-C1	[BH22]	✨

✨ = **new result**

References

[W02] David Wagner. A Generalized Birthday Problem. https://www.iacr.org/archive/crypto2002/24420288/24420288.pdf

[FOR17] Pooya Farshim, Claudio Orlandi, and Răzvan Roşie. Security of Symmetric Primitives under Incorrect Usage of Keys. ia.cr/2017/288

[GLR17] Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. Message franking via committing authenticated encryption. ia.cr/2017/664

[DGRW19] Yevgeniy Dodis, Paul Grubbs, Thomas Ristenpart, and Joanne Woodage. Fast message franking: From invisible salamanders to encryptment. ia.cr/2019/016

[LGR20] Julia Len, Paul Grubbs, and Thomas Ristenpart. Partitioning Oracle Attacks. ia.cr/2020/1491

[ADGKLS20] Ange Albertini, Thai Duong, Shay Gueron, Stefan Kölbl, Atul Luykx, and Sophie Schmieg. How to Abuse and Fix Authenticated Encryption Without Key Commitment. ia.cr/2020/1456

[BH22] Mihir Bellare and Viet Tung Hoang. Efficient schemes for committing authenticated encryption. ia.cr/2022/268

[RS07] Philip Rogaway and Thomas Shrimpton. The SIV Mode of Operation for Deterministic Authenticated-Encryption (Key Wrap) and Misuse-Resistant Nonce-Based Authenticated-Encryption. https://web.cs.ucdavis.edu/~rogaway/papers/siv.pdf