# Counter-in-Tweak: Authenticated Encryption Modes for Tweakable Block Ciphers

Thomas Peyrin[1] Yannick Seurin[2]

[1]NTU, Singapore

[2]ANSSI, France

August 15, 2016 — CRYPTO 2016

# Context

- starting point: CAESAR competition for Authenticated Encryption (AE)
- more precisely, candidates Deoxys, Joltik and KIASU (Jean, Nikolic, Peyrin)
- each is based on a tweakable block cipher (Deoxys-BC, Joltik-BC, or KIASU-BC) and two modes of operation:
  - ΘCB for the nonce-respecting setting
  - COPA for the nonce-misuse setting
- problems with COPA:
  - provides only *online* nonce-misuse resistance [FFL12, HRRV15]
  - for fractional messages, relied on XLS which has been broken [Nan14]

# Context

- starting point: CAESAR competition for Authenticated Encryption (AE)
- more precisely, candidates Deoxys, Joltik and KIASU (Jean, Nikolic, Peyrin)
- each is based on a tweakable block cipher (Deoxys-BC, Joltik-BC, or KIASU-BC) and two modes of operation:
  - ΘCB for the nonce-respecting setting
  - COPA for the nonce-misuse setting
- problems with COPA:
  - provides only *online* nonce-misuse resistance [FFL12, HRRV15]
  - for fractional messages, relied on XLS which has been broken [Nan14]

# Context

- starting point: CAESAR competition for Authenticated Encryption (AE)
- more precisely, candidates Deoxys, Joltik and KIASU (Jean, Nikolic, Peyrin)
- each is based on a tweakable block cipher (Deoxys-BC, Joltik-BC, or KIASU-BC) and two modes of operation:
  - ΘCB for the nonce-respecting setting
  - COPA for the nonce-misuse setting
- problems with COPA:
  - provides only *online* nonce-misuse resistance [FFL12, HRRV15]
  - for fractional messages, relied on XLS which has been broken [Nan14]

# Context

- starting point: CAESAR competition for Authenticated Encryption (AE)
- more precisely, candidates Deoxys, Joltik and KIASU (Jean, Nikolic, Peyrin)
- each is based on a tweakable block cipher (Deoxys-BC, Joltik-BC, or KIASU-BC) and two modes of operation:
  - ΘCB for the nonce-respecting setting
  - COPA for the nonce-misuse setting
- problems with COPA:
  - provides only *online* nonce-misuse resistance [FFL12, HRRV15]
  - for fractional messages, relied on XLS which has been broken [Nan14]

# Context

- starting point: CAESAR competition for Authenticated Encryption (AE)
- more precisely, candidates Deoxys, Joltik and KIASU (Jean, Nikolic, Peyrin)
- each is based on a tweakable block cipher (Deoxys-BC, Joltik-BC, or KIASU-BC) and two modes of operation:
  - ΘCB for the nonce-respecting setting
  - COPA for the nonce-misuse setting
- problems with COPA:
  - provides only *online* nonce-misuse resistance [FFL12, HRRV15]
  - for fractional messages, relied on XLS which has been broken [Nan14]

# Context

- starting point: CAESAR competition for Authenticated Encryption (AE)
- more precisely, candidates Deoxys, Joltik and KIASU (Jean, Nikolic, Peyrin)
- each is based on a tweakable block cipher (Deoxys-BC, Joltik-BC, or KIASU-BC) and two modes of operation:
  - ΘCB for the nonce-respecting setting
  - COPA for the nonce-misuse setting
- problems with COPA:
  - provides only *online* nonce-misuse resistance [FFL12, HRRV15]
  - for fractional messages, relied on XLS which has been broken [Nan14]

# Context

- starting point: CAESAR competition for Authenticated Encryption (AE)
- more precisely, candidates Deoxys, Joltik and KIASU (Jean, Nikolic, Peyrin)
- each is based on a tweakable block cipher (Deoxys-BC, Joltik-BC, or KIASU-BC) and two modes of operation:
  - ΘCB for the nonce-respecting setting
  - COPA for the nonce-misuse setting
- problems with COPA:
  - provides only *online* nonce-misuse resistance [FFL12, HRRV15]
  - for fractional messages, relied on XLS which has been broken [Nan14]

# Context

- starting point: CAESAR competition for Authenticated Encryption (AE)
- more precisely, candidates Deoxys, Joltik and KIASU (Jean, Nikolic, Peyrin)
- each is based on a tweakable block cipher (Deoxys-BC, Joltik-BC, or KIASU-BC) and two modes of operation:
  - $\Theta$CB for the nonce-respecting setting
  - COPA for the nonce-misuse setting
- problems with COPA:
  - provides only *online* nonce-misuse resistance [FFL12, HRRV15]
  - for fractional messages, relied on XLS which has been broken [Nan14]

# Our Goal

- in replacement of COPA, design an AE mode of operation for tweakable block ciphers which provides:
  1. (full, not online) nonce-misuse resistance up to the birthday bound
  2. beyond-birthday-bound (BBB) security in the nonce-respecting setting
- existing (TBC $\Rightarrow$ AE) modes:
  - ΘCB [KR11] is perfectly secure in the nonce-respecting scenario, but not secure at all in the nonce-misuse scenario
  - COPA [ABL+13] provides only *online* nonce-misuse resistance
  - AEZ [HKR15] provides birthday-security even in the nonce-respecting scenario
  - PIV [ST13] requires a very long tweak-length (size of the maximal message length)
- our new mode = SCT (*Synthetic Counter in Tweak*)

# Our Goal

- in replacement of COPA, design an AE mode of operation for tweakable block ciphers which provides:
  1. (full, not online) nonce-misuse resistance up to the birthday bound
  2. beyond-birthday-bound (BBB) security in the nonce-respecting setting

- existing (TBC $\Rightarrow$ AE) modes:
  - ΘCB [KR11] is perfectly secure in the nonce-respecting scenario, but not secure at all in the nonce-misuse scenario
  - COPA [ABL+13] provides only *online* nonce-misuse resistance
  - AEZ [HKR15] provides birthday-security even in the nonce-respecting scenario
  - PIV [ST13] requires a very long tweak-length (size of the maximal message length)

- our new mode = SCT (*Synthetic Counter in Tweak*)

# Our Goal

- in replacement of COPA, design an AE mode of operation for tweakable block ciphers which provides:
  1. (full, not online) nonce-misuse resistance up to the birthday bound
  2. beyond-birthday-bound (BBB) security in the nonce-respecting setting

- existing (TBC $\Rightarrow$ AE) modes:
  - ΘCB [KR11] is perfectly secure in the nonce-respecting scenario, but not secure at all in the nonce-misuse scenario
  - COPA [ABL+13] provides only *online* nonce-misuse resistance
  - AEZ [HKR15] provides birthday-security even in the nonce-respecting scenario
  - PIV [ST13] requires a very long tweak-length (size of the maximal message length)

- our new mode = SCT (*Synthetic Counter in Tweak*)

# Our Goal

- in replacement of COPA, design an AE mode of operation for tweakable block ciphers which provides:
    1. (full, not online) nonce-misuse resistance up to the birthday bound
    2. beyond-birthday-bound (BBB) security in the nonce-respecting setting

- existing (TBC $\Rightarrow$ AE) modes:
    - ΘCB [KR11] is perfectly secure in the nonce-respecting scenario, but not secure at all in the nonce-misuse scenario
    - COPA [ABL+13] provides only *online* nonce-misuse resistance
    - AEZ [HKR15] provides birthday-security even in the nonce-respecting scenario
    - PIV [ST13] requires a very long tweak-length (size of the maximal message length)
- our new mode = SCT (*Synthetic Counter in Tweak*)

# Our Goal

- in replacement of COPA, design an AE mode of operation for tweakable block ciphers which provides:
    1. (full, not online) nonce-misuse resistance up to the birthday bound
    2. beyond-birthday-bound (BBB) security in the nonce-respecting setting

- existing (TBC $\Rightarrow$ AE) modes:
    - ΘCB [KR11] is perfectly secure in the nonce-respecting scenario, but not secure at all in the nonce-misuse scenario
    - COPA [ABL+13] provides only *online* nonce-misuse resistance
    - AEZ [HKR15] provides birthday-security even in the nonce-respecting scenario
    - PIV [ST13] requires a very long tweak-length (size of the maximal message length)

- our new mode = SCT (*Synthetic Counter in Tweak*)

# Our Goal

- in replacement of COPA, design an AE mode of operation for tweakable block ciphers which provides:
  1. (full, not online) nonce-misuse resistance up to the birthday bound
  2. beyond-birthday-bound (BBB) security in the nonce-respecting setting

- existing (TBC $\Rightarrow$ AE) modes:
  - ΘCB [KR11] is perfectly secure in the nonce-respecting scenario, but not secure at all in the nonce-misuse scenario
  - COPA [ABL+13] provides only *online* nonce-misuse resistance
  - AEZ [HKR15] provides birthday-security even in the nonce-respecting scenario
  - PIV [ST13] requires a very long tweak-length (size of the maximal message length)

- our new mode = SCT (*Synthetic Counter in Tweak*)

# Our Goal

- in replacement of COPA, design an AE mode of operation for tweakable block ciphers which provides:
  1. (full, not online) nonce-misuse resistance up to the birthday bound
  2. beyond-birthday-bound (BBB) security in the nonce-respecting setting

- existing (TBC $\Rightarrow$ AE) modes:
  - ΘCB [KR11] is perfectly secure in the nonce-respecting scenario, but not secure at all in the nonce-misuse scenario
  - COPA [ABL$^+$13] provides only *online* nonce-misuse resistance
  - AEZ [HKR15] provides birthday-security even in the nonce-respecting scenario
  - PIV [ST13] requires a very long tweak-length (size of the maximal message length)

- our new mode = SCT (*Synthetic Counter in Tweak*)

# Our Goal

- in replacement of COPA, design an AE mode of operation for tweakable block ciphers which provides:
  1. (full, not online) nonce-misuse resistance up to the birthday bound
  2. beyond-birthday-bound (BBB) security in the nonce-respecting setting

- existing (TBC $\Rightarrow$ AE) modes:
  - ΘCB [KR11] is perfectly secure in the nonce-respecting scenario, but not secure at all in the nonce-misuse scenario
  - COPA [ABL+13] provides only *online* nonce-misuse resistance
  - AEZ [HKR15] provides birthday-security even in the nonce-respecting scenario
  - PIV [ST13] requires a very long tweak-length (size of the maximal message length)

- our new mode = SCT (*Synthetic Counter in Tweak*)

# Our Goal

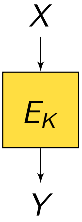
- in replacement of COPA, design an AE mode of operation for tweakable block ciphers which provides:
  1. (full, not online) nonce-misuse resistance up to the birthday bound
  2. beyond-birthday-bound (BBB) security in the nonce-respecting setting

- existing (TBC $\Rightarrow$ AE) modes:
  - ΘCB [KR11] is perfectly secure in the nonce-respecting scenario, but not secure at all in the nonce-misuse scenario
  - COPA [ABL+13] provides only *online* nonce-misuse resistance
  - AEZ [HKR15] provides birthday-security even in the nonce-respecting scenario
  - PIV [ST13] requires a very long tweak-length (size of the maximal message length)
- our new mode = SCT (*Synthetic Counter in Tweak*)

# Outline

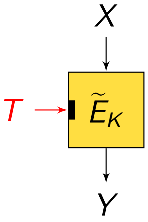
# Outline

# Building Block: Tweakable Block Ciphers (TBCs)



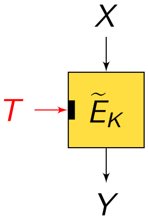

- tweak $T$: brings variability to the block cipher
- $T$ assumed public or even adversarially controlled
- each tweak should give an "independent" permutation
- few "natively tweakable" BCs:
  - Hasty Pudding Cipher [Sch98]
  - Mercy [Cro00]
  - Threefish [FLS+10]
  - CAESAR proposals KIASU, Deoxys, Joltik, (i)SCREAM, Minalpher

# Building Block: Tweakable Block Ciphers (TBCs)



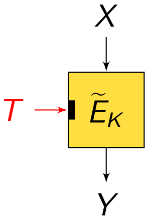

- tweak $T$: brings variability to the block cipher
- $T$ assumed public or even adversarially controlled
- each tweak should give an "independent" permutation
- few "natively tweakable" BCs:
  - Hasty Pudding Cipher [Sch98]
  - Mercy [Cro00]
  - Threefish [FLS+10]
  - CAESAR proposals KIASU, Deoxys, Joltik, (i)SCREAM, Minalpher

# Building Block: Tweakable Block Ciphers (TBCs)



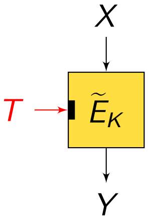

- tweak $T$: brings variability to the block cipher
- $T$ assumed public or even adversarially controlled
- each tweak should give an "independent" permutation
- few "natively tweakable" BCs:
  - Hasty Pudding Cipher [Sch98]
  - Mercy [Cro00]
  - Threefish [FLS+10]
  - CAESAR proposals KIASU, Deoxys, Joltik, (i)SCREAM, Minalpher

# Building Block: Tweakable Block Ciphers (TBCs)



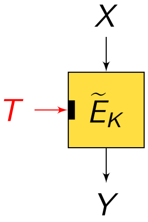

- tweak $T$: brings variability to the block cipher
- $T$ assumed public or even adversarially controlled
- each tweak should give an "independent" permutation
- few "natively tweakable" BCs:
  - Hasty Pudding Cipher [Sch98]
  - Mercy [Cro00]
  - Threefish [FLS+10]
  - CAESAR proposals KIASU, Deoxys, Joltik, (i)SCREAM, Minalpher

# Building Block: Tweakable Block Ciphers (TBCs)



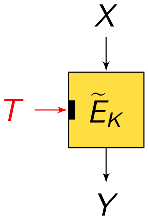

- tweak $T$: brings variability to the block cipher
- $T$ assumed public or even adversarially controlled
- each tweak should give an "independent" permutation
- few "natively tweakable" BCs:
  - Hasty Pudding Cipher [Sch98]
  - Mercy [Cro00]
  - Threefish [FLS+10]
  - CAESAR proposals KIASU, Deoxys, Joltik, (i)SCREAM, Minalpher

# Building Block: Tweakable Block Ciphers (TBCs)



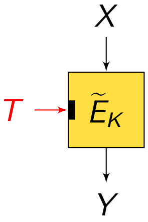

- tweak $T$: brings variability to the block cipher
- $T$ assumed public or even adversarially controlled
- each tweak should give an "independent" permutation
- few "natively tweakable" BCs:
  - Hasty Pudding Cipher [Sch98]
  - Mercy [Cro00]
  - Threefish [FLS+10]
  - CAESAR proposals KIASU, Deoxys, Joltik, (i)SCREAM, Minalpher

# Building Block: Tweakable Block Ciphers (TBCs)



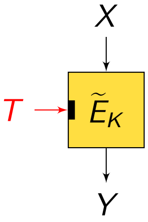

- tweak $T$: brings variability to the block cipher
- $T$ assumed public or even adversarially controlled
- each tweak should give an "independent" permutation
- few "natively tweakable" BCs:
  - Hasty Pudding Cipher [Sch98]
  - Mercy [Cro00]
  - Threefish [FLS+10]
  - CAESAR proposals KIASU, Deoxys, Joltik, (i)SCREAM, Minalpher

# Building Block: Tweakable Block Ciphers (TBCs)




- tweak $T$: brings variability to the block cipher
- $T$ assumed public or even adversarially controlled
- each tweak should give an "independent" permutation
- few "natively tweakable" BCs:
  - Hasty Pudding Cipher [Sch98]
  - Mercy [Cro00]
  - Threefish [FLS+10]
  - CAESAR proposals KIASU, Deoxys, Joltik, (i)SCREAM, Minalpher

# Building Block: Tweakable Block Ciphers (TBCs)




- tweak $T$: brings variability to the block cipher
- $T$ assumed public or even adversarially controlled
- each tweak should give an "independent" permutation
- few "natively tweakable" BCs:
  - Hasty Pudding Cipher [Sch98]
  - Mercy [Cro00]
  - Threefish [FLS+10]
  - CAESAR proposals KIASU, Deoxys, Joltik, (i)SCREAM, Minalpher

# Goal: Nonce-Based Authenticated Encryption (nAE)

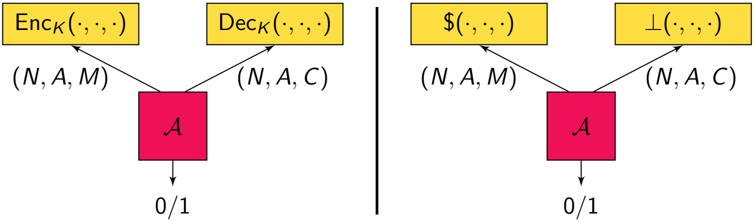
## Syntax

A nAE scheme $\Pi$ is a pair of algorithms $(\Pi.\mathrm{Enc}, \Pi.\mathrm{Dec})$ where

- algorithm $\Pi.\mathrm{Enc}$ takes
  - (a key $K$)
  - a nonce $N$
  - associated data $A$
  - a message $M$

  and returns a ciphertext $C$.
- algorithm $\Pi.\mathrm{Dec}$ takes $K$ and $(N, A, C)$ and returns $M$ or $\perp$.

# Goal: Nonce-Based Authenticated Encryption (nAE)




## Security (all-in-one definition)

- The scheme $\Pi$ is secure if adversary $\mathcal{A}$ cannot distinguish $(\mathrm{Enc}_K, \mathrm{Dec}_K)$ and $(\$, \bot)$.
- $\mathcal{A}$ cannot ask a decryption query $(N, A, C)$ if it received $C$ from an encryption query $(N, A, M)$
- $\mathcal{A}$ is said nonce-respecting if it never repeats a nonce in encryption queries.

# Misuse-Resistant AE (MRAE)

## Nonce-misuse resistance (informal) [RS06]

A nAE scheme is said nonce-misuse resistant if repeating a nonce in encryption queries:

- does not harm authenticity
- hurts confidentiality only insofar as repetitions of triplets $(N, A, M)$ are detectable

- $\simeq$ deterministic authenticated encryption
- MRAE schemes *cannot* be online (each ciphertext bit must depend on each input bit)

# Misuse-Resistant AE (MRAE)

## Nonce-misuse resistance (informal) [RS06]

A nAE scheme is said nonce-misuse resistant if repeating a nonce in encryption queries:

- does not harm authenticity
- hurts confidentiality only insofar as repetitions of triplets $(N, A, M)$ are detectable

- $\simeq$ deterministic authenticated encryption
- MRAE schemes *cannot* be online (each ciphertext bit must depend on each input bit)

# Misuse-Resistant AE (MRAE)

## Nonce-misuse resistance (informal) [RS06]

A nAE scheme is said nonce-misuse resistant if repeating a nonce in encryption queries:

- does not harm authenticity
- hurts confidentiality only insofar as repetitions of triplets $(N, A, M)$ are detectable

- $\simeq$ deterministic authenticated encryption
- MRAE schemes *cannot* be online (each ciphertext bit must depend on each input bit)

# Outline

# Generic Composition

Starting from two building blocks:

- a MAC (or a PRF) $F_{K_1}(\cdot, \cdot, \cdot)$
- an encryption scheme $\text{Enc}_{K_2}(\cdot, \cdot)$

combine them to obtain a nAE scheme [BN00, NRS14].

Two types of encryption schemes:

- (random) IV-based encryption (ivE): $C = \text{Enc}_{K_2}(IV, M)$, $IV$ randomly chosen by the encryption oracle (ex: CBC)
- nonce-based encryption (nE): $C = \text{Enc}_{K_2}(N, M)$, $N$ chosen by the adversary but non-repeating (ex: nonce-based CTR mode, GCM)

# Generic Composition

Starting from two building blocks:

- a MAC (or a PRF) $F_{K_1}(\cdot, \cdot, \cdot)$
- an encryption scheme $\mathsf{Enc}_{K_2}(\cdot, \cdot)$

combine them to obtain a nAE scheme [BN00, NRS14].

Two types of encryption schemes:

- (random) IV-based encryption (ivE): $C = \mathsf{Enc}_{K_2}(IV, M)$, $IV$ randomly chosen by the encryption oracle (ex: CBC)
- nonce-based encryption (nE): $C = \mathsf{Enc}_{K_2}(N, M)$, $N$ chosen by the adversary but non-repeating (ex: nonce-based CTR mode, GCM)

# Generic Composition

Starting from two building blocks:

- a MAC (or a PRF) $F_{K_1}(\cdot,\cdot,\cdot)$
- an encryption scheme $\mathrm{Enc}_{K_2}(\cdot,\cdot)$

combine them to obtain a nAE scheme [BN00, NRS14].

Two types of encryption schemes:

- (random) IV-based encryption (ivE):
  $C = \mathrm{Enc}_{K_2}(IV, M)$, $IV$ randomly chosen by the encryption oracle (ex: CBC)
- nonce-based encryption (nE):
  $C = \mathrm{Enc}_{K_2}(N, M)$, $N$ chosen by the adversary but non-repeating (ex: nonce-based CTR mode, GCM)

# Generic Composition

Starting from two building blocks:

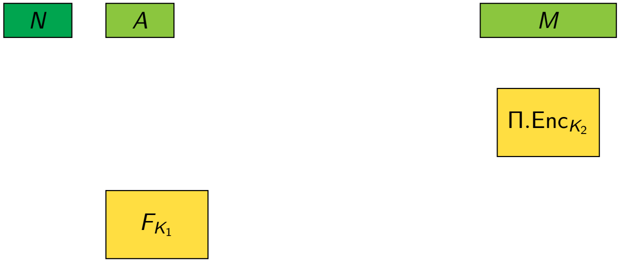
- a MAC (or a PRF) $F_{K_1}(\cdot,\cdot,\cdot)$
- an encryption scheme $\mathrm{Enc}_{K_2}(\cdot,\cdot)$

combine them to obtain a nAE scheme [BN00, NRS14].

Two types of encryption schemes:

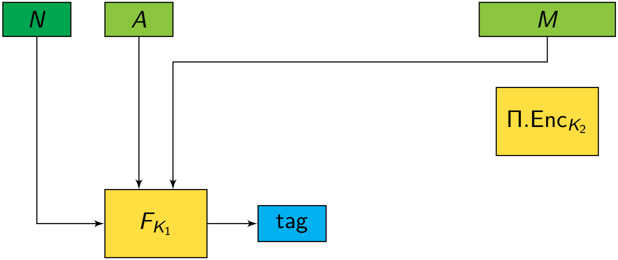
- (random) IV-based encryption (ivE):
  $C = \mathrm{Enc}_{K_2}(IV, M)$, $IV$ randomly chosen by the encryption oracle (ex: CBC)
- nonce-based encryption (nE):
  $C = \mathrm{Enc}_{K_2}(N, M)$, $N$ chosen by the adversary but non-repeating (ex: nonce-based CTR mode, GCM)

# From SIV to NSIV



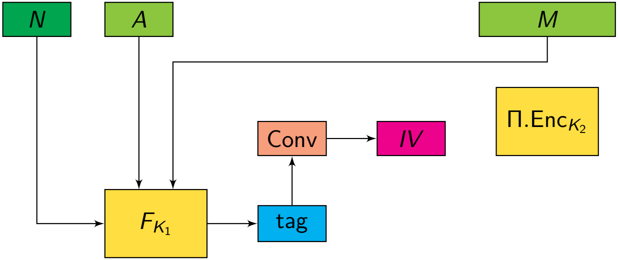

- SIV (*Synthetic IV*) [RS06] combines a PRF $F_{K_1}(N, A, M)$ and an IV-based encryption scheme $\Pi.\mathrm{Enc}_{K_2}(IV, M)$
- provides nonce-misuse resistance up to the birthday-bound from birthday-secure components (e.g. CMAC + CTR encryption)
- what about BBB-security in the nonce-respecting case?

# From SIV to NSIV



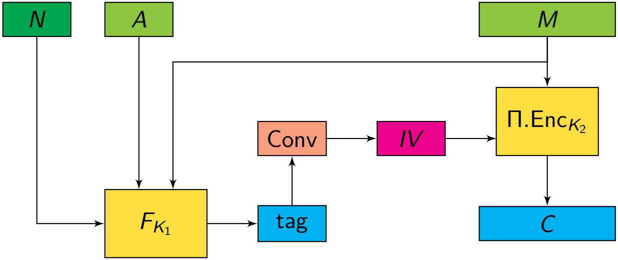

- SIV (*Synthetic IV*) [RS06] combines a PRF $F_{K_1}(N, A, M)$ and an IV-based encryption scheme $\Pi.\text{Enc}_{K_2}(IV, M)$
- provides nonce-misuse resistance up to the birthday-bound from birthday-secure components (e.g. CMAC + CTR encryption)
- what about BBB-security in the nonce-respecting case?

# From SIV to NSIV



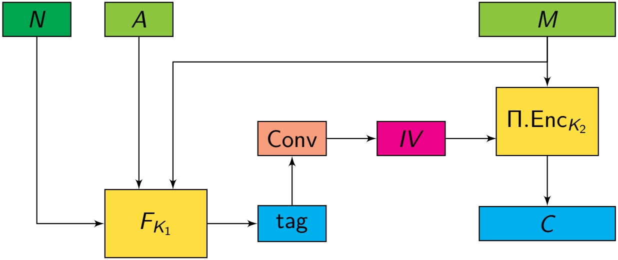

- SIV (*Synthetic IV*) [RS06] combines a PRF $F_{K_1}(N, A, M)$ and an IV-based encryption scheme $\Pi.\mathrm{Enc}_{K_2}(IV, M)$
- provides nonce-misuse resistance up to the birthday-bound from birthday-secure components (e.g. CMAC + CTR encryption)
- what about BBB-security in the nonce-respecting case?

# From SIV to NSIV



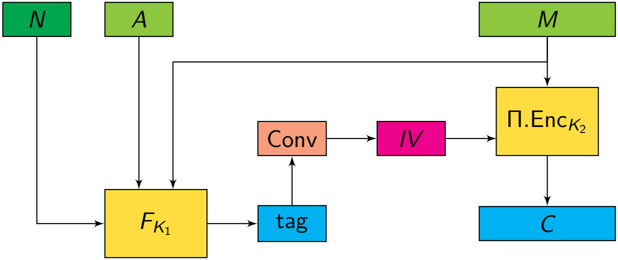

- SIV (*Synthetic IV*) [RS06] combines a PRF $F_{K_1}(N, A, M)$ and an IV-based encryption scheme $\Pi.\mathrm{Enc}_{K_2}(IV, M)$
- provides nonce-misuse resistance up to the birthday-bound from birthday-secure components (e.g. CMAC + CTR encryption)
- what about BBB-security in the nonce-respecting case?

# From SIV to NSIV



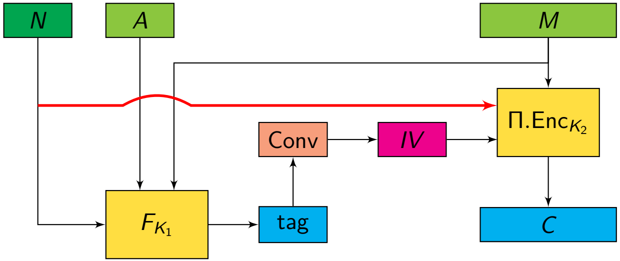

- SIV (*Synthetic IV*) [RS06] combines a PRF $F_{K_1}(N, A, M)$ and an IV-based encryption scheme $\Pi.\mathrm{Enc}_{K_2}(IV, M)$
- provides nonce-misuse resistance up to the birthday-bound from birthday-secure components (e.g. CMAC + CTR encryption)
- what about BBB-security in the nonce-respecting case?

# From SIV to NSIV



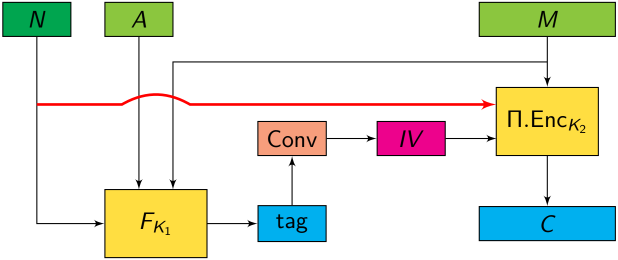

- SIV (*Synthetic IV*) [RS06] combines a PRF $F_{K_1}(N, A, M)$ and an IV-based encryption scheme $\Pi.\mathrm{Enc}_{K_2}(IV, M)$
- provides nonce-misuse resistance up to the birthday-bound from birthday-secure components (e.g. CMAC + CTR encryption)
- what about BBB-security in the nonce-respecting case?

# From SIV to NSIV



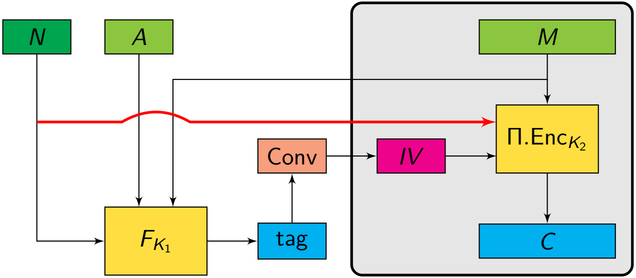

- SIV (*Synthetic IV*) [RS06] combines a PRF $F_{K_1}(N, A, M)$ and an IV-based encryption scheme $\Pi.\mathrm{Enc}_{K_2}(IV, M)$
- provides nonce-misuse resistance up to the birthday-bound from birthday-secure components (e.g. CMAC + CTR encryption)
- what about BBB-security in the nonce-respecting case?
  $\Rightarrow$ Re-use the nonce $N$ in the encryption scheme!

# Combined Nonce and IV-based (nivE) Encryption



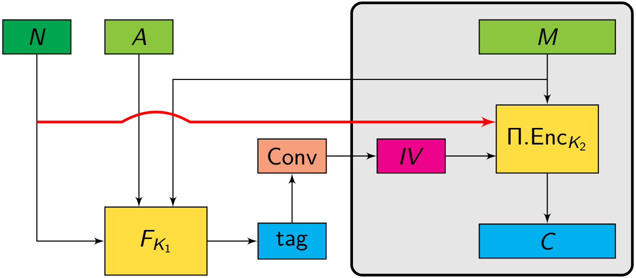

- the encryption algorithm Π.Enc takes a nonce and a random IV!
- security definition: ciphertexts must be indist. from random, assuming nonces do not repeat and IV is random
- NB: when nonces can be repeated, ≃ (family of) standard IV-based encryption scheme

# Combined Nonce and IV-based (nivE) Encryption



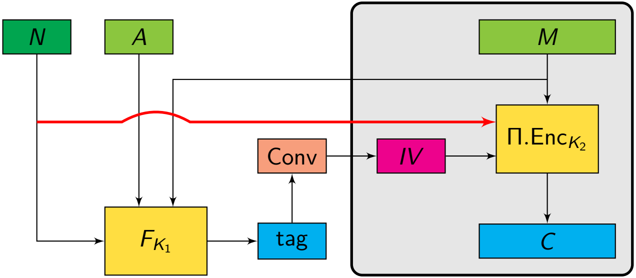

- the encryption algorithm Π.Enc takes a nonce and a random IV!
- security definition: ciphertexts must be indist. from random, assuming nonces do not repeat and IV is random
- NB: when nonces can be repeated, ≃ (family of) standard IV-based encryption scheme

# Combined Nonce and IV-based (nivE) Encryption



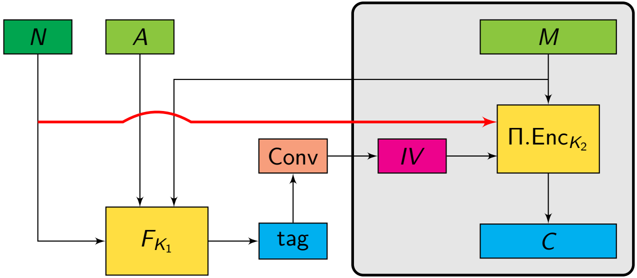

- the encryption algorithm Π.Enc takes a nonce and a random IV!
- security definition: ciphertexts must be indist. from random, assuming nonces do not repeat and IV is random
- NB: when nonces can be repeated, ≃ (family of) standard IV-based encryption scheme

# Combined Nonce and IV-based (nivE) Encryption



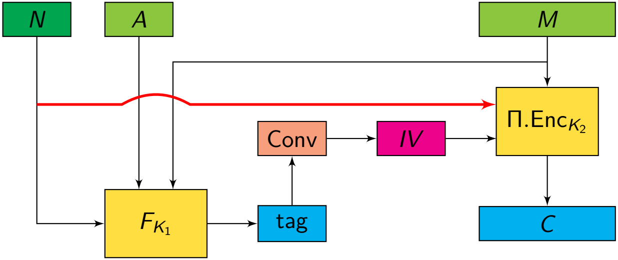

- the encryption algorithm Π.Enc takes a nonce and a random IV!
- security definition: ciphertexts must be indist. from random, assuming nonces do not repeat and IV is random
- NB: when nonces can be repeated, ≃ (family of) standard IV-based encryption scheme

# Combined Nonce and IV-based (nivE) Encryption




- the encryption algorithm Π.Enc takes a nonce and a random IV!
- security definition: ciphertexts must be indist. from random, assuming nonces do not repeat and IV is random
- NB: when nonces can be repeated, ≃ (family of) standard IV-based encryption scheme

# Security Result for NSIV



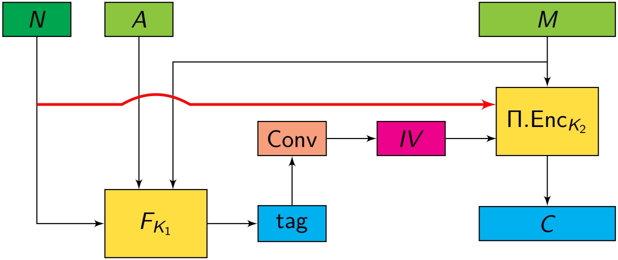

## Theorem

*For any adversary* $\mathcal{A}$ *against* NSIV$[F, \Pi]$,

$$\mathbf{Adv}^{\mathrm{nAE}}_{\mathsf{NSIV}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{nivE}}_{\Pi}(\mathcal{A}') + \mathbf{Adv}^{\mathrm{nPRF}}_{F}(\mathcal{A}'') + \mathbf{Adv}^{\mathrm{nMAC}}_{F}(\mathcal{A}''').$$

*Moreover, if* $\mathcal{A}$ *repeats any nonce at most m times, then* $\mathcal{A}'$, $\mathcal{A}''$, *and* $\mathcal{A}'''$ *also repeat any nonce at most m times.*

# Instantiating $F$ and $\Pi$




## Remaining of the talk:

How to instantiate the PRF $F$ and the nivE encryption scheme $\Pi$ from a TBC $\widetilde{E}$ so that

- we get BBB-security in the nonce-respecting setting
- we retain birthday-bound security in the nonce-misuse setting

# Outline

TBCs and AE

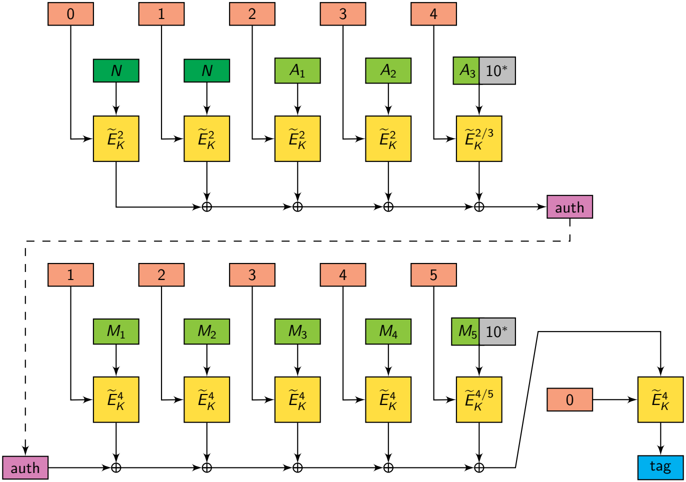
Generic Composition: the NSIV Method

Authentication: the EPWC Mode

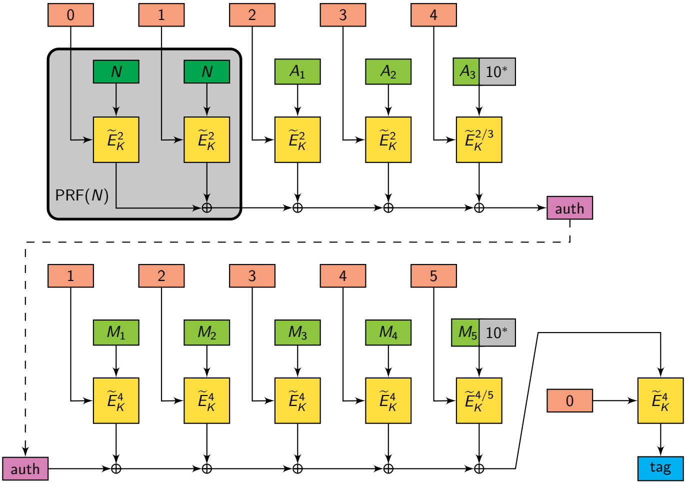
Encryption: the CTRT Mode

Conclusion

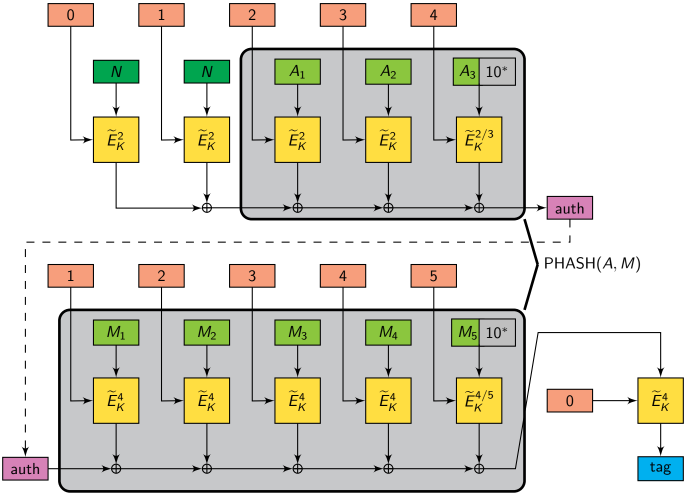
# The EPWC (*Encrypted Parallel Wegman-Carter*) Mode



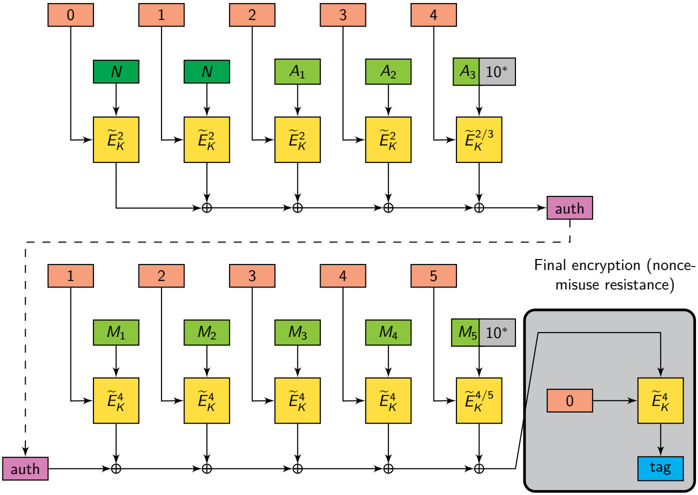
# The EPWC (*Encrypted Parallel Wegman-Carter*) Mode



# The EPWC (*Encrypted Parallel Wegman-Carter*) Mode



# The EPWC (*Encrypted Parallel Wegman-Carter*) Mode



# Security of EPWC

**Theorem**

*Let $\mathcal{A}$ be an adversary against* EPWC *with an ideal TBC with block-length $n$ making at most $q$ queries. Then*

(a) *If $\mathcal{A}$ is nonce-respecting,*

$$\mathbf{Adv}^{\mathrm{nPRF}}_{\mathsf{EPWC}}(\mathcal{A}) \leq \mathcal{O}\left(\frac{q}{2^n}\right), \qquad \mathbf{Adv}^{\mathrm{nMAC}}_{\mathsf{EPWC}}(\mathcal{A}) \leq \mathcal{O}\left(\frac{q}{2^n}\right).$$

(b) *If $\mathcal{A}$ is allowed to repeat nonces, then*

$$\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{EPWC}}(\mathcal{A}) \leq \frac{q^2}{2^n}, \qquad \mathbf{Adv}^{\mathrm{MAC}}_{\mathsf{EPWC}}(\mathcal{A}) \leq \frac{q^2+q}{2^n}.$$

# Outline

TBCs and AE

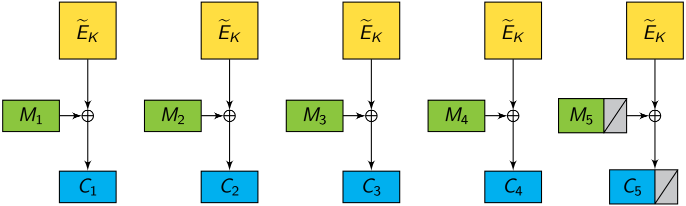
Generic Composition: the NSIV Method

Authentication: the EPWC Mode

Encryption: the CTRT Mode

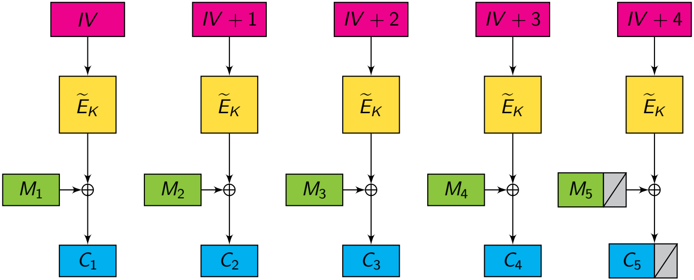
Conclusion

# The CTRT (*CounTeR-in-Tweak*) Encryption Mode



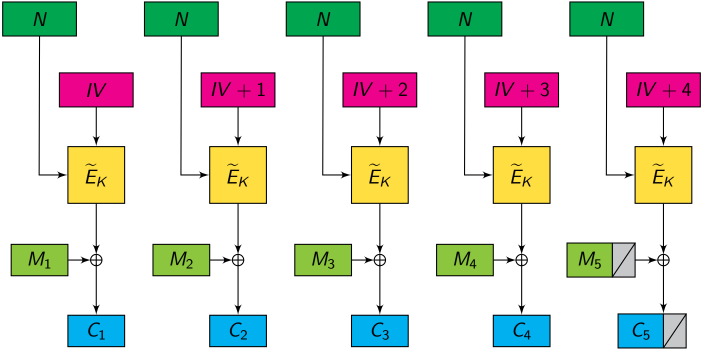

- how to build a counter-like nivE encryption scheme?
- nonce in the tweak ⇒ birthday attack!
- switch inputs: nonce in "message input" and counter in tweak
- key observation: $T \mapsto \widetilde{E}_K(T, N)$ is a pseudorandom *function*

# The CTRT (*CounTeR-in-Tweak*) Encryption Mode



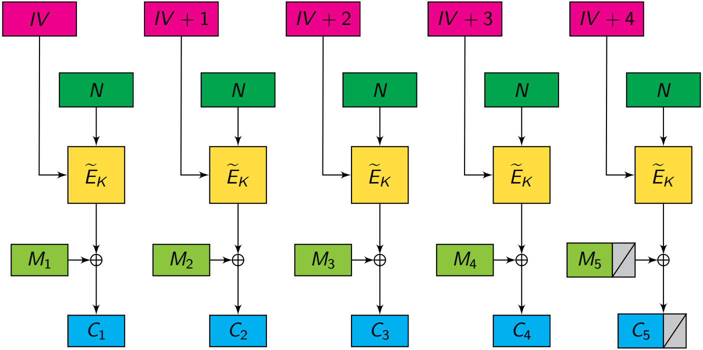

- how to build a counter-like nivE encryption scheme?
- nonce in the tweak $\Rightarrow$ birthday attack!
- switch inputs: nonce in "message input" and counter in tweak
- key observation: $T \mapsto \widetilde{E}_K(T, N)$ is a pseudorandom *function*

# The CTRT (*CounTeR-in-Tweak*) Encryption Mode



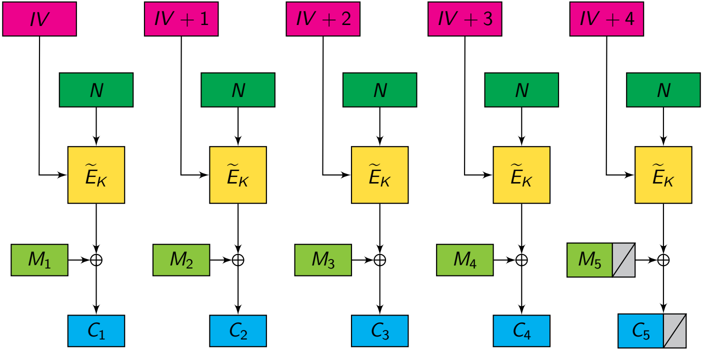

- how to build a counter-like nivE encryption scheme?
- nonce in the tweak ⇒ birthday attack!
- switch inputs: nonce in "message input" and counter in tweak
- key observation: $T \mapsto \widetilde{E}_K(T, N)$ is a pseudorandom *function*

# The CTRT (*CounTeR-in-Tweak*) Encryption Mode




- how to build a counter-like nivE encryption scheme?
- nonce in the tweak $\Rightarrow$ birthday attack!
- switch inputs: nonce in “message input” and counter in tweak
- key observation: $T \mapsto \widetilde{E}_K(T, N)$ is a pseudorandom *function*

# The CTRT (*CounTeR-in-Tweak*) Encryption Mode




- how to build a counter-like nivE encryption scheme?
- nonce in the tweak $\Rightarrow$ birthday attack!
- switch inputs: nonce in "message input" and counter in tweak
- key observation: $T \mapsto \widetilde{E}_K(T, N)$ is a pseudorandom *function*

# Security of CTRT

## Theorem

- *$n$ = block-length*
- *$t$ = tweak-length*
- *$\sigma$ = total length of queries (in n-bit blocks)*
- *$m$ = maximal number of repetitions of any nonce*

$$\mathbf{Adv}^{\mathrm{nivE}}_{\mathsf{CTRT}}(\mathcal{A}) \leq \frac{2(m-1)\sigma}{2^t} + \frac{1}{2^t} + \begin{cases} \dfrac{2\sigma\log^2\sigma}{2^n} & \text{when } \sigma \leq 2^t, \\ \dfrac{2t^2\sigma^2}{2^{n+t}} & \text{when } \sigma \geq 2^t. \end{cases}$$

- nonce-respecting ($m = 1$): security up to $\sigma \simeq \min\{2^n, 2^{(n+t)/2}\}$
- security degrades "gracefully" with the maximal number of nonce repetitions $m$

# Security of CTRT

## Theorem

- $n =$ *block-length*
- $t =$ *tweak-length*
- $\sigma =$ *total length of queries (in n-bit blocks)*
- $m =$ *maximal number of repetitions of any nonce*

$$\mathbf{Adv}^{\mathrm{nivE}}_{\mathsf{CTRT}}(\mathcal{A}) \leq \frac{2(m-1)\sigma}{2^t} + \frac{1}{2^t} + \begin{cases} \dfrac{2\sigma\log^2\sigma}{2^n} & \text{when } \sigma \leq 2^t, \\ \dfrac{2t^2\sigma^2}{2^{n+t}} & \text{when } \sigma \geq 2^t. \end{cases}$$

- nonce-respecting ($m = 1$): security up to $\sigma \simeq \min\{2^n, 2^{(n+t)/2}\}$
- security degrades "gracefully" with the maximal number of nonce repetitions $m$

# Security of CTRT

## Theorem

- $n =$ *block-length*
- $t =$ *tweak-length*
- $\sigma =$ *total length of queries (in n-bit blocks)*
- $m =$ *maximal number of repetitions of any nonce*

$$\mathbf{Adv}^{\mathrm{nivE}}_{\mathsf{CTRT}}(\mathcal{A}) \leq \begin{cases} \frac{1}{2^t} + \frac{2\sigma \log^2 \sigma}{2^n} & \text{when } \sigma \leq 2^t, \\ + \frac{2t^2\sigma^2}{2^{n+t}} & \text{when } \sigma \geq 2^t. \end{cases}$$

- nonce-respecting ($m = 1$): security up to $\sigma \simeq \min\{2^n, 2^{(n+t)/2}\}$
- security degrades "gracefully" with the maximal number of nonce repetitions $m$

# Security of CTRT

## Theorem

- $n =$ *block-length*
- $t =$ *tweak-length*
- $\sigma =$ *total length of queries (in n-bit blocks)*
- $m =$ *maximal number of repetitions of any nonce*

$$\mathbf{Adv}^{\mathrm{nivE}}_{\mathsf{CTRT}}(\mathcal{A}) \leq \begin{cases} \dfrac{1}{2^t} + \dfrac{2\sigma \log^2 \sigma}{2^n} & \textit{when } \sigma \leq 2^t, \\ + \dfrac{2t^2\sigma^2}{2^{n+t}} & \textit{when } \sigma \geq 2^t. \end{cases}$$

- nonce-respecting ($m = 1$): security up to $\sigma \simeq \min\{2^n, 2^{(n+t)/2}\}$
- security degrades "gracefully" with the maximal number of nonce repetitions $m$

# Security of CTRT

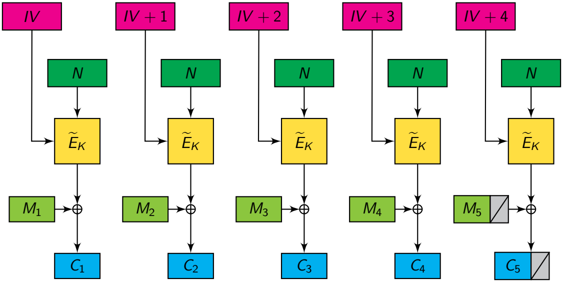
## Theorem

- *$n =$ block-length*
- *$t =$ tweak-length*
- *$\sigma =$ total length of queries (in n-bit blocks)*
- *$m =$ maximal number of repetitions of any nonce*

$$\mathbf{Adv}^{\mathrm{nivE}}_{\mathrm{CTRT}}(\mathcal{A}) \le \frac{2(m-1)\sigma}{2^t} + \frac{1}{2^t} + \begin{cases} \dfrac{2\sigma\log^2\sigma}{2^n} & \text{when } \sigma \le 2^t, \\ \dfrac{2t^2\sigma^2}{2^{n+t}} & \text{when } \sigma \ge 2^t. \end{cases}$$

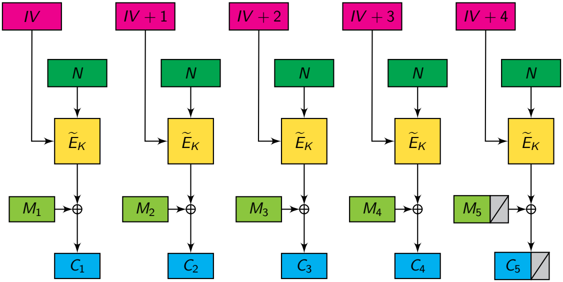
- nonce-respecting ($m = 1$): security up to $\sigma \simeq \min\{2^n, 2^{(n+t)/2}\}$
- security degrades "gracefully" with the maximal number of nonce repetitions $m$

# Proof of Security of CTRT (nonce-respecting)



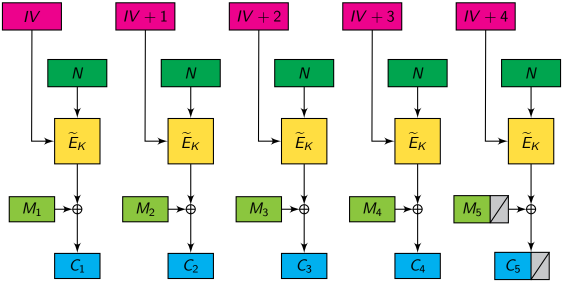

- assume first that nonces are never repeated
- we want to show that ciphertexts are indist. from random
- each random IV determines the sequence of tweaks $(IV, IV+1, \ldots)$ used in the TBC
- for each tweak $T \in \mathcal{T}$, let $L(T)$ ("load") be the number of times the tweak $T$ has been used throughout encryption queries

# Proof of Security of CTRT (nonce-respecting)



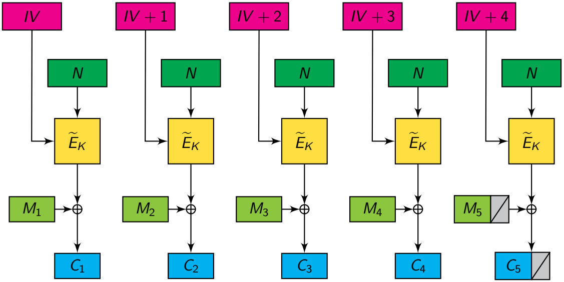

- assume first that nonces are never repeated
- we want to show that ciphertexts are indist. from random
- each random IV determines the sequence of tweaks $(IV, IV + 1, \ldots)$ used in the TBC
- for each tweak $T \in \mathcal{T}$, let $L(T)$ ("load") be the number of times the tweak $T$ has been used throughout encryption queries

# Proof of Security of CTRT (nonce-respecting)



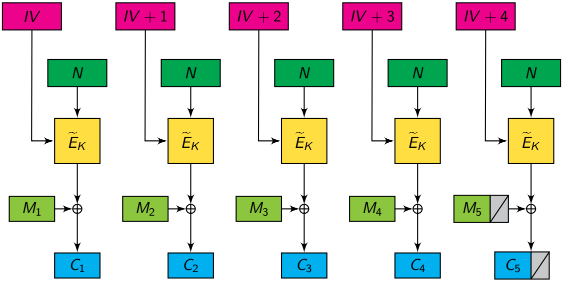

- assume first that nonces are never repeated
- we want to show that ciphertexts are indist. from random
- each random IV determines the sequence of tweaks $(IV, IV+1, \ldots)$ used in the TBC
- for each tweak $T \in \mathcal{T}$, let $L(T)$ ("load") be the number of times the tweak $T$ has been used throughout encryption queries

# Proof of Security of CTRT (nonce-respecting)




- assume first that nonces are never repeated
- we want to show that ciphertexts are indist. from random
- each random IV determines the sequence of tweaks $(IV, IV+1, \ldots)$ used in the TBC
- for each tweak $T \in \mathcal{T}$, let $L(T)$ ("load") be the number of times the tweak $T$ has been used throughout encryption queries

# Proof of Security of CTRT (nonce-respecting)

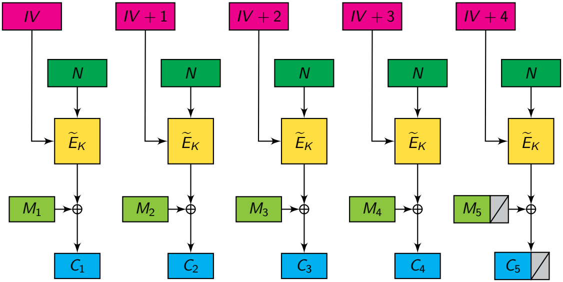



- for each tweak, we have an independent PRF/PRP distinguishing problem with $L(T)$ "queries" (nonces):

$$\mathbf{Adv}(\mathcal{A}) \leq \sum_{T \in \mathcal{T}} \frac{L(T)^2}{2 \cdot 2^n} \leq \min\{\sigma, 2^t\} \cdot \frac{(L_{\max})^2}{2 \cdot 2^n}$$

- upper bound on $L_{\max} = \max L(T)$: "balls-into-bins" problem

# Proof of Security of CTRT (nonce-respecting)

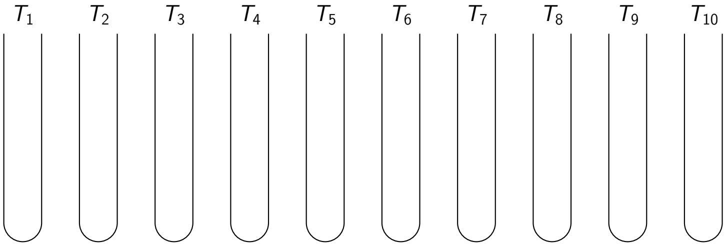



- for each tweak, we have an independent PRF/PRP distinguishing problem with $L(T)$ "queries" (nonces):

$$\mathbf{Adv}(\mathcal{A}) \leq \sum_{T \in \mathcal{T}} \frac{L(T)^2}{2 \cdot 2^n} \leq \min\{\sigma, 2^t\} \cdot \frac{(L_{\max})^2}{2 \cdot 2^n}$$

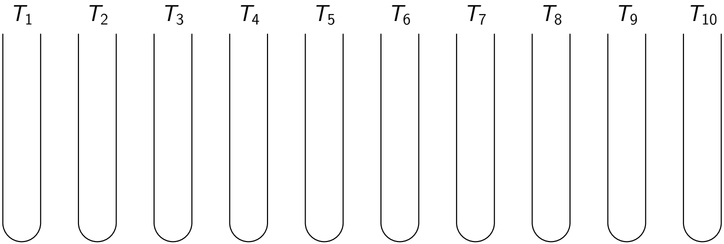
- upper bound on $L_{\max} = \max L(T)$: "balls-into-bins" problem

# Proof of Security of CTRT (nonce-respecting)



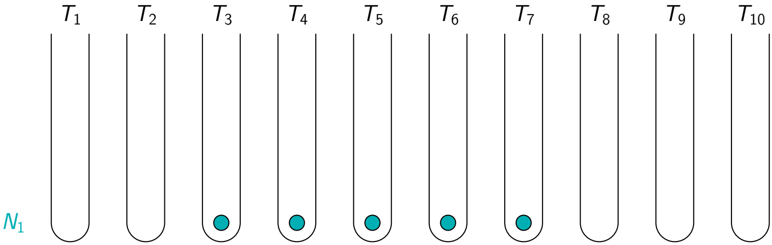

- $2^t$ bins = tweak values
- $\sigma$ balls = nonces
- for each query, the random IV determines in which (consecutive) bins the nonces are thrown
- except with probability $1/2^t$, one has
  - (a) if $\sigma \leq 2^t$, then $\max L(T) \leq 2 \log \sigma$;
  - (b) if $\sigma \geq 2^t$, then $\max L(T) \leq \frac{2t\sigma}{2^t}$.

# Proof of Security of CTRT (nonce-respecting)



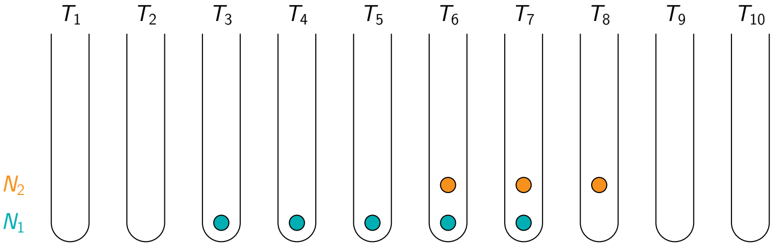

- $2^t$ bins = tweak values
- $\sigma$ balls = nonces
- for each query, the random IV determines in which (consecutive) bins the nonces are thrown
- except with probability $1/2^t$, one has
  - (a) if $\sigma \leq 2^t$, then $\max L(T) \leq 2 \log \sigma$;
  - (b) if $\sigma \geq 2^t$, then $\max L(T) \leq \frac{2t\sigma}{2^t}$.

# Proof of Security of CTRT (nonce-respecting)



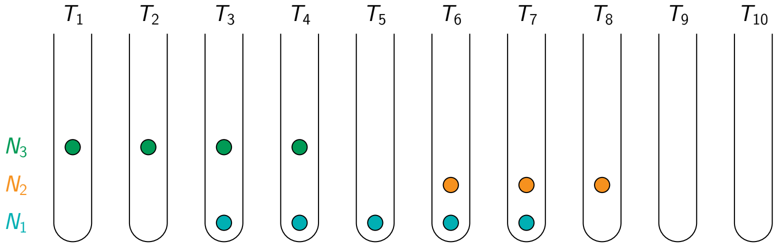

- $2^t$ bins = tweak values
- $\sigma$ balls = nonces
- for each query, the random IV determines in which (consecutive) bins the nonces are thrown
- except with probability $1/2^t$, one has
  - (a) if $\sigma \leq 2^t$, then $\max L(T) \leq 2 \log \sigma$;
  - (b) if $\sigma \geq 2^t$, then $\max L(T) \leq \frac{2t\sigma}{2^t}$.

# Proof of Security of CTRT (nonce-respecting)



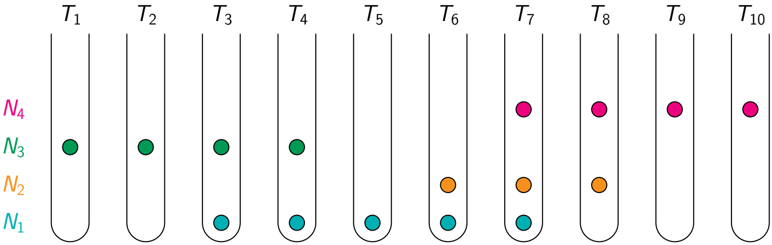

- $2^t$ bins = tweak values
- $\sigma$ balls = nonces
- for each query, the random IV determines in which (consecutive) bins the nonces are thrown
- except with probability $1/2^t$, one has
  - (a) if $\sigma \leq 2^t$, then $\max L(T) \leq 2 \log \sigma$;
  - (b) if $\sigma \geq 2^t$, then $\max L(T) \leq \frac{2t\sigma}{2^t}$.

# Proof of Security of CTRT (nonce-respecting)



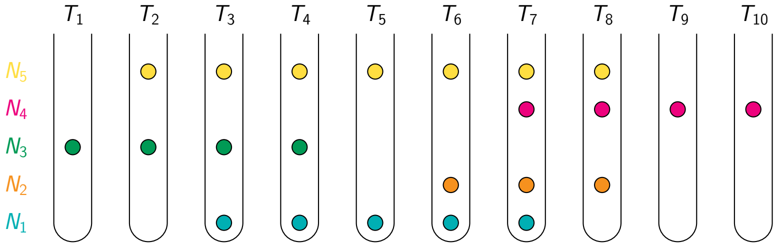

- $2^t$ bins = tweak values
- $\sigma$ balls = nonces
- for each query, the random IV determines in which (consecutive) bins the nonces are thrown
- except with probability $1/2^t$, one has
  - (a) if $\sigma \leq 2^t$, then $\max L(T) \leq 2 \log \sigma$;
  - (b) if $\sigma \geq 2^t$, then $\max L(T) \leq \frac{2t\sigma}{2^t}$.

# Proof of Security of CTRT (nonce-respecting)



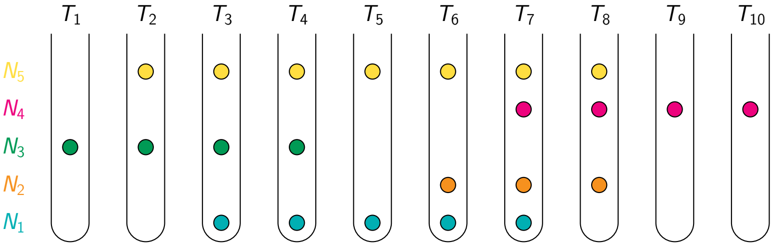

- $2^t$ bins = tweak values
- $\sigma$ balls = nonces
- for each query, the random IV determines in which (consecutive) bins the nonces are thrown
- except with probability $1/2^t$, one has
  (a) if $\sigma \leq 2^t$, then $\max L(T) \leq 2 \log \sigma$;
  (b) if $\sigma \geq 2^t$, then $\max L(T) \leq \frac{2t\sigma}{2^t}$.

# Proof of Security of CTRT (nonce-respecting)



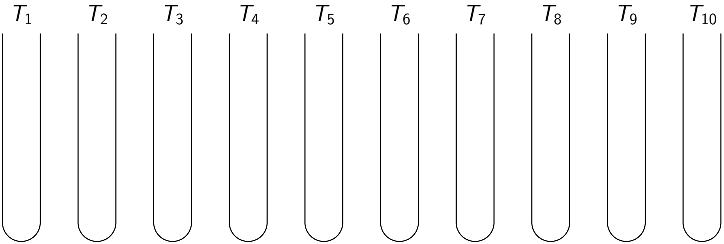

- $2^t$ bins = tweak values
- $\sigma$ balls = nonces
- for each query, the random IV determines in which (consecutive) bins the nonces are thrown
- except with probability $1/2^t$, one has
  - (a) if $\sigma \le 2^t$, then $\max L(T) \le 2 \log \sigma$;
  - (b) if $\sigma \ge 2^t$, then $\max L(T) \le \frac{2t\sigma}{2^t}$.

# Proof of Security of CTRT (nonce-respecting)



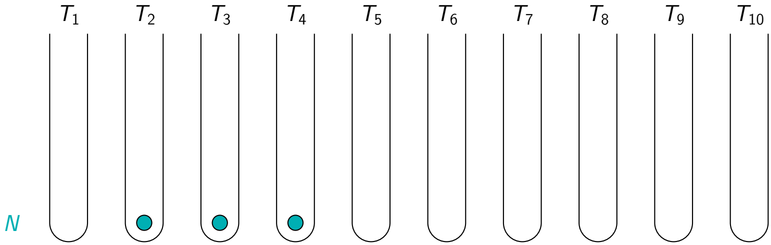

- $2^t$ bins = tweak values
- $\sigma$ balls = nonces
- for each query, the random IV determines in which (consecutive) bins the nonces are thrown
- except with probability $1/2^t$, one has
  - (a) if $\sigma \leq 2^t$, then $\max L(T) \leq 2 \log \sigma$;
  - (b) if $\sigma \geq 2^t$, then $\max L(T) \leq \frac{2t\sigma}{2^t}$.

# Proof of Security of CTRT (nonce-misuse)



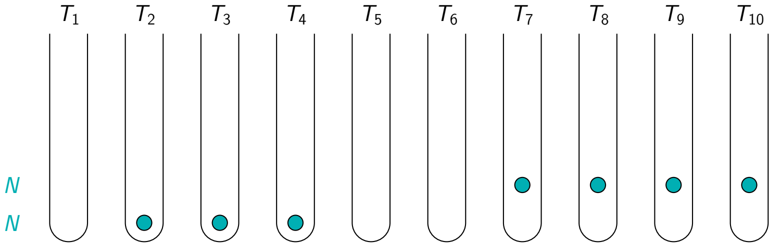

- bad event that allows to distinguish outputs from random: $\exists$ two encryption queries with the same nonce and a common tweak (counter)
- for two messages of length $\ell$ and $\ell'$, happens with proba. $(\ell + \ell' - 1)/2^t$
- yields the term $(m-1)\sigma/2^t$ in the security bound

# Proof of Security of CTRT (nonce-misuse)



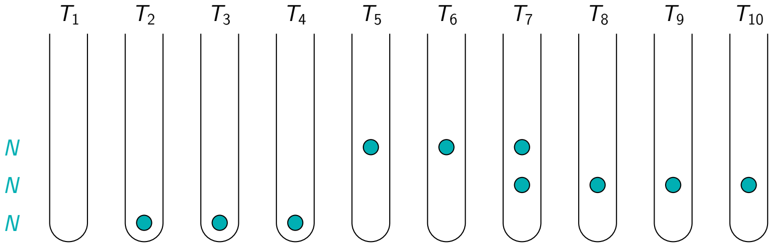

- bad event that allows to distinguish outputs from random: $\exists$ two encryption queries with the same nonce and a common tweak (counter)
- for two messages of length $\ell$ and $\ell'$, happens with proba. $(\ell + \ell' - 1)/2^t$
- yields the term $(m-1)\sigma/2^t$ in the security bound

# Proof of Security of CTRT (nonce-misuse)



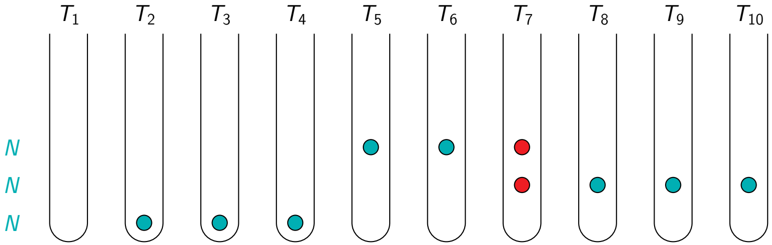

- bad event that allows to distinguish outputs from random: $\exists$ two encryption queries with the same nonce and a common tweak (counter)
- for two messages of length $\ell$ and $\ell'$, happens with proba. $(\ell + \ell' - 1)/2^t$
- yields the term $(m-1)\sigma/2^t$ in the security bound

# Proof of Security of CTRT (nonce-misuse)



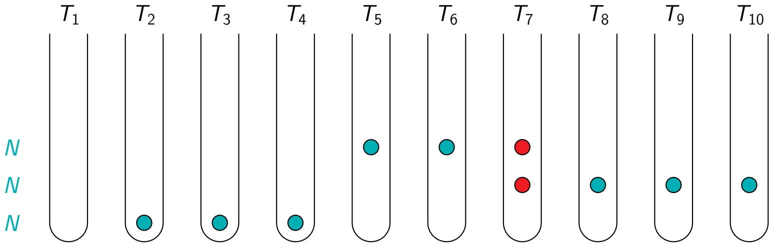

- bad event that allows to distinguish outputs from random: $\exists$ two encryption queries with the same nonce and a common tweak (counter)
- for two messages of length $\ell$ and $\ell'$, happens with proba. $(\ell + \ell' - 1)/2^t$
- yields the term $(m-1)\sigma/2^t$ in the security bound

# Proof of Security of CTRT (nonce-misuse)



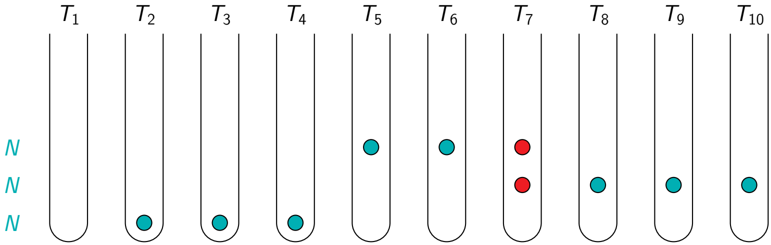

- bad event that allows to distinguish outputs from random: $\exists$ two encryption queries with the same nonce and a common tweak (counter)
- for two messages of length $\ell$ and $\ell'$, happens with proba. $(\ell + \ell' - 1)/2^t$
- yields the term $(m-1)\sigma/2^t$ in the security bound

# Proof of Security of CTRT (nonce-misuse)




- bad event that allows to distinguish outputs from random: $\exists$ two encryption queries with the same nonce and a common tweak (counter)
- for two messages of length $\ell$ and $\ell'$, happens with proba. $(\ell + \ell' - 1)/2^t$
- yields the term $(m-1)\sigma/2^t$ in the security bound

# Proof of Security of CTRT (nonce-misuse)




- bad event that allows to distinguish outputs from random: $\exists$ two encryption queries with the same nonce and a common tweak (counter)
- for two messages of length $\ell$ and $\ell'$, happens with proba. $(\ell + \ell' - 1)/2^t$
- yields the term $(m-1)\sigma/2^t$ in the security bound

# Outline

# Wrap-up and Final Remarks

- EPWC + CTRT combined using the NSIV composition method
  = SCT (*Synthetic Counter in Tweak*) mode
- BBB-secure in the nonce-respecting setting
- retains birthday-bound security in the nonce-misuse setting
- parallel, quite efficient, does not need the decryption direction
- instantiation of the TBC: needs to be BBB-secure!
  ⇒ XEX does not work
  ⇒ use ad-hoc TBCs such as Deoxys-BC and Joltik-BC

# Wrap-up and Final Remarks

- EPWC + CTRT combined using the NSIV composition method
  = SCT (*Synthetic Counter in Tweak*) mode
- BBB-secure in the nonce-respecting setting
- retains birthday-bound security in the nonce-misuse setting
- parallel, quite efficient, does not need the decryption direction
- instantiation of the TBC: needs to be BBB-secure!
  $\Rightarrow$ XEX does not work
  $\Rightarrow$ use ad-hoc TBCs such as Deoxys-BC and Joltik-BC

# Wrap-up and Final Remarks

- EPWC + CTRT combined using the NSIV composition method
  = SCT (*Synthetic Counter in Tweak*) mode
- BBB-secure in the nonce-respecting setting
- retains birthday-bound security in the nonce-misuse setting
- parallel, quite efficient, does not need the decryption direction
- instantiation of the TBC: needs to be BBB-secure!
  ⇒ XEX does not work
  ⇒ use ad-hoc TBCs such as Deoxys-BC and Joltik-BC

# Wrap-up and Final Remarks

- EPWC + CTRT combined using the NSIV composition method
  = SCT (*Synthetic Counter in Tweak*) mode
- BBB-secure in the nonce-respecting setting
- retains birthday-bound security in the nonce-misuse setting
- parallel, quite efficient, does not need the decryption direction
- instantiation of the TBC: needs to be BBB-secure!
  ⇒ XEX does not work
  ⇒ use ad-hoc TBCs such as Deoxys-BC and Joltik-BC

# Wrap-up and Final Remarks

- EPWC + CTRT combined using the NSIV composition method
  = SCT (*Synthetic Counter in Tweak*) mode
- BBB-secure in the nonce-respecting setting
- retains birthday-bound security in the nonce-misuse setting
- parallel, quite efficient, does not need the decryption direction
- instantiation of the TBC: needs to be BBB-secure!
  ⇒ XEX does not work
  ⇒ use ad-hoc TBCs such as Deoxys-BC and Joltik-BC

# The end...

Thanks for your attention!

Comments or questions?

# References I


- Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Elmar Tischhauser, and Kan Yasuda. Parallelizable and Authenticated Online Ciphers. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 (Proceedings, Part I)*, volume 8269 of *LNCS*, pages 424–443. Springer, 2013.
- Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 531–545. Springer, 2000.
- Paul Crowley. Mercy: A Fast Large Block Cipher for Disk Sector Encryption. In Bruce Schneier, editor, *Fast Software Encryption - FSE 2000*, volume 1978 of *LNCS*, pages 49–63. Springer, 2000.
- Ewan Fleischmann, Christian Forler, and Stefan Lucks. McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. In Anne Canteaut, editor, *Fast Software Encryption - FSE 2012*, volume 7549 of *LNCS*, pages 196–215. Springer, 2012.

# References II


- Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker. The Skein Hash Function Family. SHA3 Submission to NIST (Round 3), 2010.

- Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust Authenticated-Encryption: AEZ and the Problem That It Solves. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 (Proceedings, Part I)*, volume 9056 of *LNCS*, pages 15–44. Springer, 2015.

- Viet Tung Hoang, Reza Reyhanitabar, Phillip Rogaway, and Damian Vizár. Online Authenticated-Encryption and its Nonce-Reuse Misuse-Resistance. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 (Proceedings, Part I)*, volume 9215 of *LNCS*, pages 493–517. Springer, 2015.

- Ted Krovetz and Phillip Rogaway. The Software Performance of Authenticated-Encryption Modes. In Antoine Joux, editor, *Fast Software Encryption - FSE 2011*, volume 6733 of *LNCS*, pages 306–327. Springer, 2011.

# References III


- Mridul Nandi. XLS is not a strong pseudorandom permutation. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 (Proceedings, Part I)*, volume 8873 of *LNCS*, pages 478–490. Springer, 2014.
- Chanathip Namprempre, Phillip Rogaway, and Thomas Shrimpton. Reconsidering Generic Composition. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 257–274. Springer, 2014.
- Phillip Rogaway and Thomas Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 373–390. Springer, 2006.
- Richard Schroeppel. The Hasty Pudding Cipher. AES submission to NIST, 1998.

# References IV

Thomas Shrimpton and R. Seth Terashima. A Modular Framework for Building Variable-Input-Length Tweakable Ciphers. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 (Proceedings, Part I)*, volume 8269 of *LNCS*, pages 405–423. Springer, 2013.