ASIACRYPT 2023

Populating the Zoo of Rugged Pseudorandom Permutations

Jean Paul Degabriele [1]

Vukašin Karadžić [2]

[1] Technology Innovation Institute, UAE

[2] Technische Universität Darmstadt, Germany

Background on Rugged Pseudorandom Permutations and their Applications

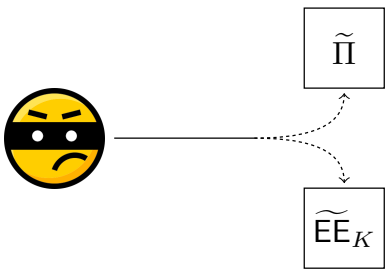
Rugged Pseudorandom Permutation (RPRP)

- Security notion for variable-length tweakable ciphers [DK22].

Rugged Pseudorandom Permutation (RPRP)

- Security notion for variable-length tweakable ciphers [DK22].

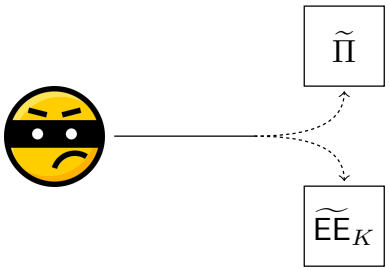
PRP



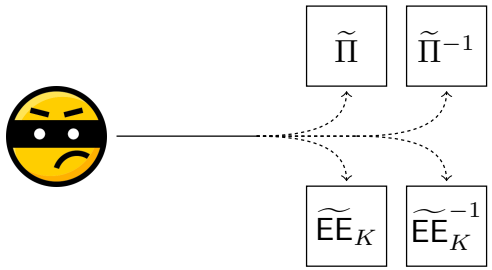
Rugged Pseudorandom Permutation (RPRP)

- Security notion for variable-length tweakable ciphers [DK22].

PRP



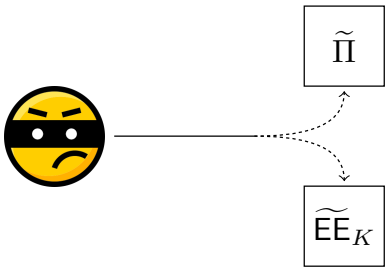
SPRP



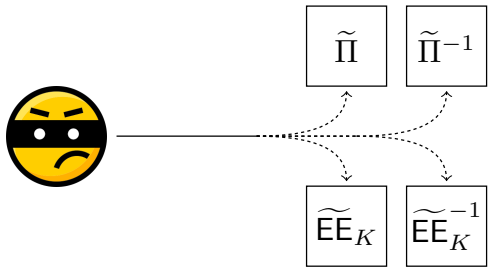
Rugged Pseudorandom Permutation (RPRP)

- Security notion for variable-length tweakable ciphers [DK22].

PRP



SPRP

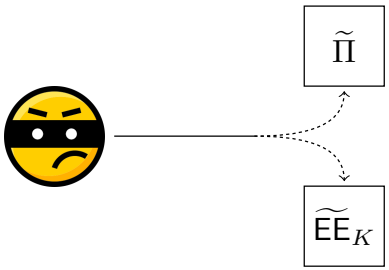


*PRP and SPRP denote the tweakable versions of the notions

Rugged Pseudorandom Permutation (RPRP)

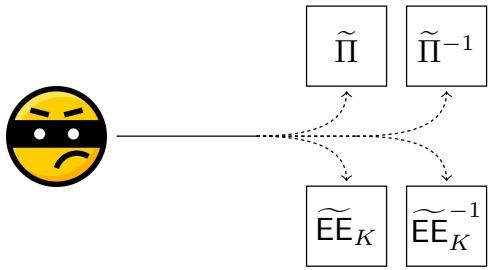
- Security notion for variable-length tweakable ciphers [DK22].

PRP



RPRP

SPRP

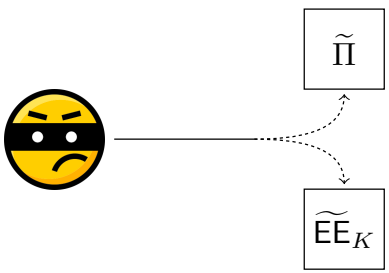


*PRP and SPRP denote the tweakable versions of the notions

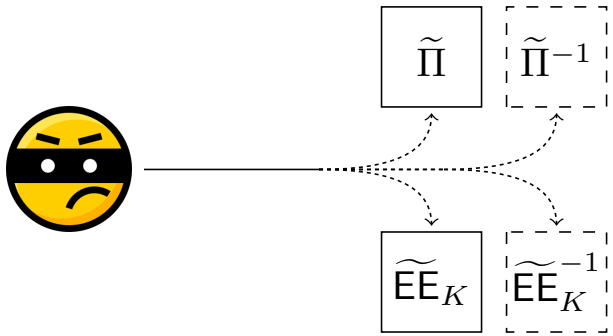
Rugged Pseudorandom Permutation (RPRP)

- Security notion for variable-length tweakable ciphers [DK22].

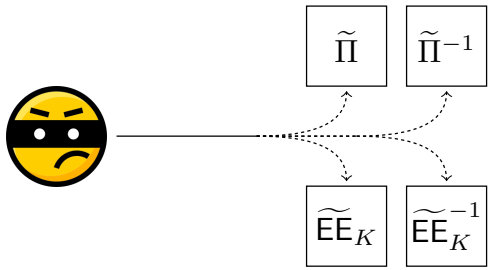
PRP



RPRP



SPRP



*PRP and SPRP denote the tweakable versions of the notions

RPRP Definition

RPRP Definition

- **Syntax**: RPRP is a variable-length tweakable cipher over a split domain $\{0,1\}^n \times \{0,1\}^*$

RPRP Definition

- **Syntax**: RPRP is a variable-length tweakable cipher over a split domain $\{0,1\}^n \times \{0,1\}^*$:

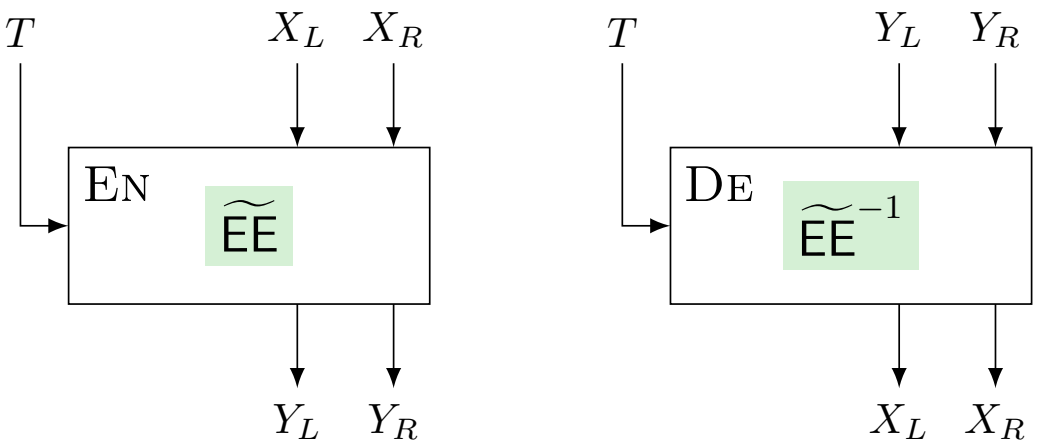
$$(Y_L, Y_R) \leftarrow \widetilde{\mathsf{EE}}(T, X_L, X_R)$$

RPRP Definition

- **Syntax**: RPRP is a variable-length tweakable cipher over a split domain $\{0,1\}^n \times \{0,1\}^*$:

$$(Y_L, Y_R) \leftarrow \widetilde{\mathsf{EE}}(T, X_L, X_R)$$

Real world

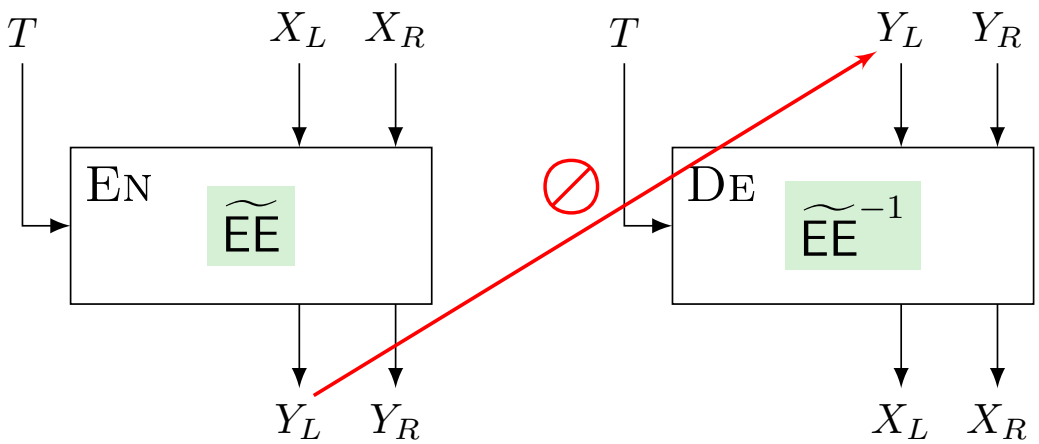


RPRP Definition

- **Syntax**: RPRP is a variable-length tweakable cipher over a split domain $\{0,1\}^n \times \{0,1\}^*$:

$$(Y_L, Y_R) \leftarrow \widetilde{\mathsf{EE}}(T, X_L, X_R)$$

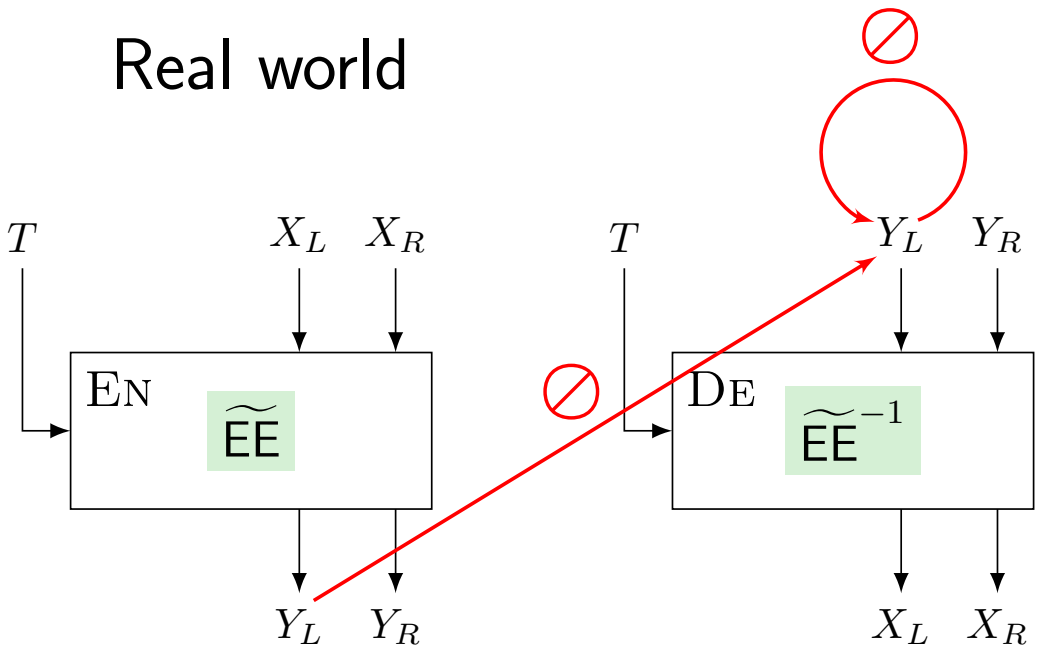
Real world



RPRP Definition

- **Syntax**: RPRP is a variable-length tweakable cipher over a split domain $\{0,1\}^n \times \{0,1\}^*$:

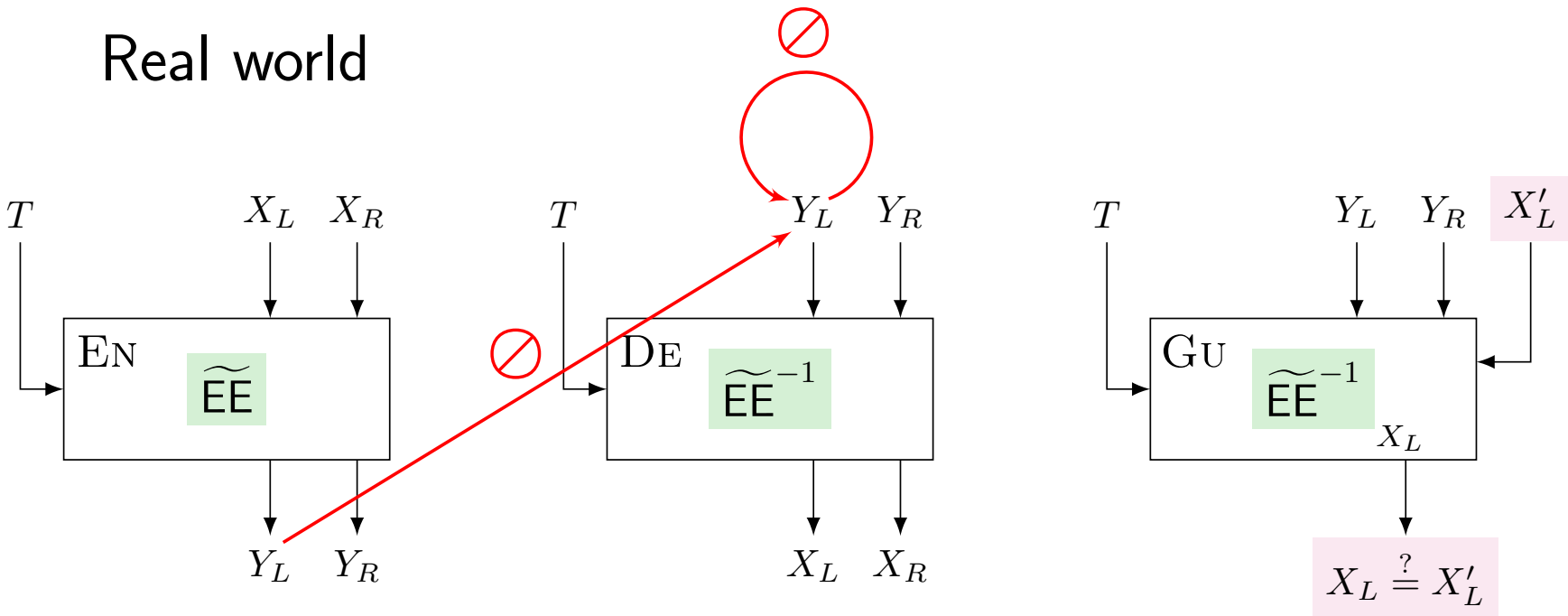
$$(Y_L, Y_R) \leftarrow \widetilde{\mathsf{EE}}(T, X_L, X_R)$$



RPRP Definition

- **Syntax**: RPRP is a variable-length tweakable cipher over a split domain $\{0,1\}^n \times \{0,1\}^*$:

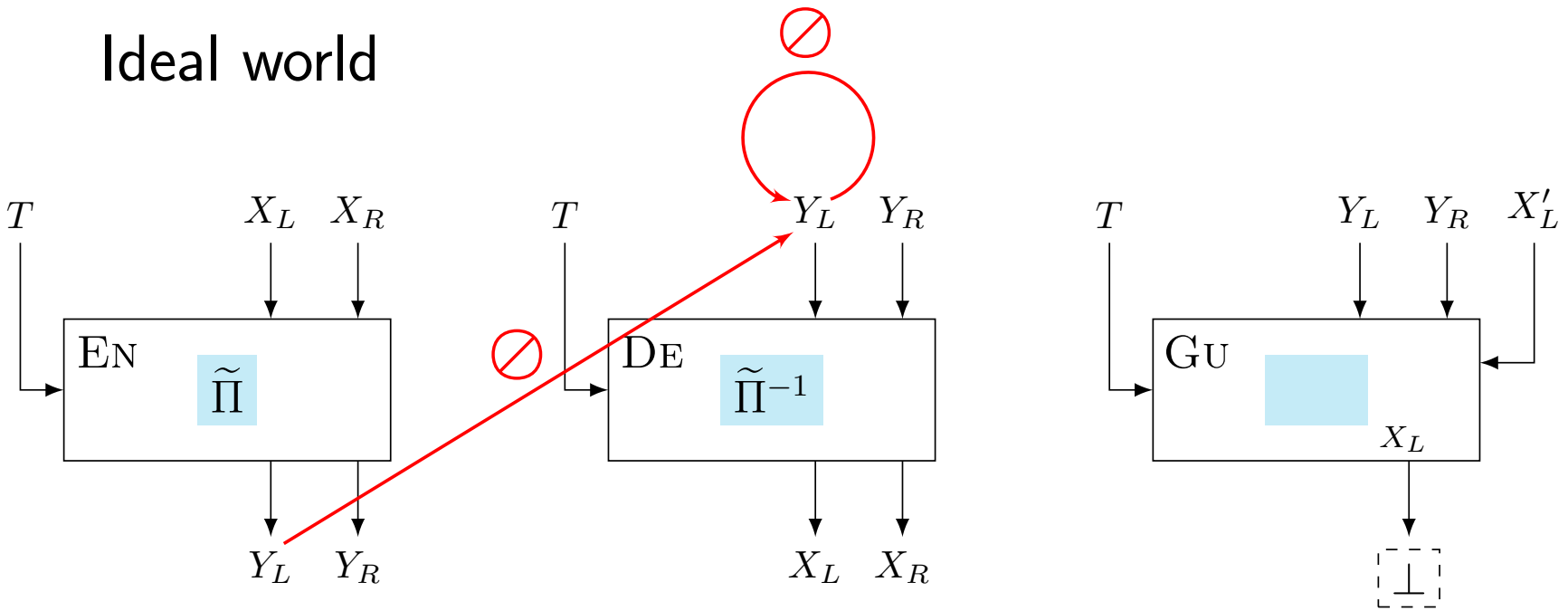
$$(Y_L, Y_R) \leftarrow \widetilde{\mathsf{EE}}(T, X_L, X_R)$$



RPRP Definition

- **Syntax**: RPRP is a variable-length tweakable cipher over a split domain $\{0,1\}^n \times \{0,1\}^*$:

$$(Y_L, Y_R) \leftarrow \widetilde{\mathsf{EE}}(T, X_L, X_R)$$



RPRP Definition

- **Syntax**: RPRP is a variable-length tweakable cipher over a split domain $\{0,1\}^n \times \{0,1\}^*$:

$$(Y_L, Y_R) \leftarrow \widetilde{\mathsf{EE}}(T, X_L, X_R)$$

RPRP Definition

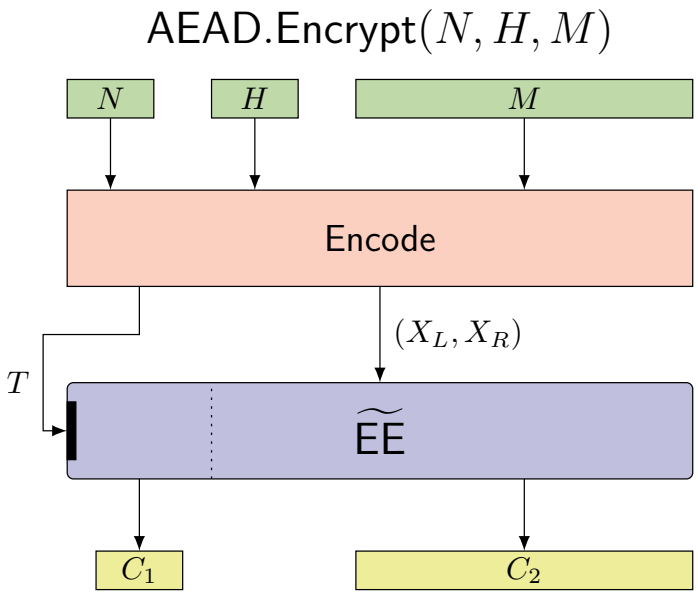
✓ More efficient than SPRP, but still **very** useful

RPRP Applications: AEAD Schemes [DK22]

- Encode-then-Encipher paradigm [BR00] with ~~SPRP~~ RPRP!

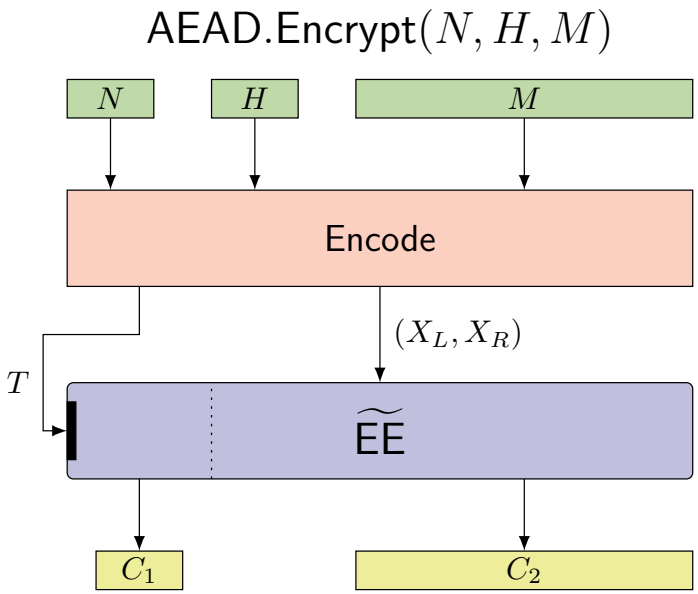
RPRP Applications: AEAD Schemes [DK22]

- Encode-then-Encipher paradigm [BR00] with ~~SPRP~~ RPRP!



RPRP Applications: AEAD Schemes [DK22]

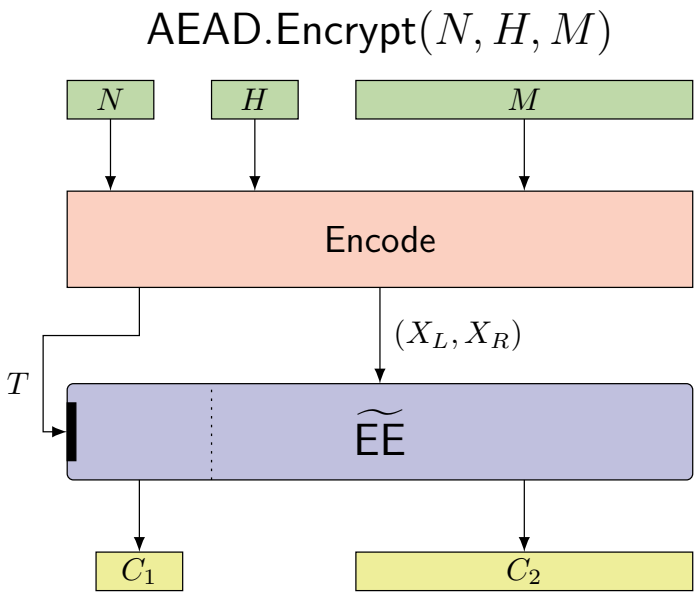
- Encode-then-Encipher paradigm [BR00] with ~~SPRP~~ RPRP!

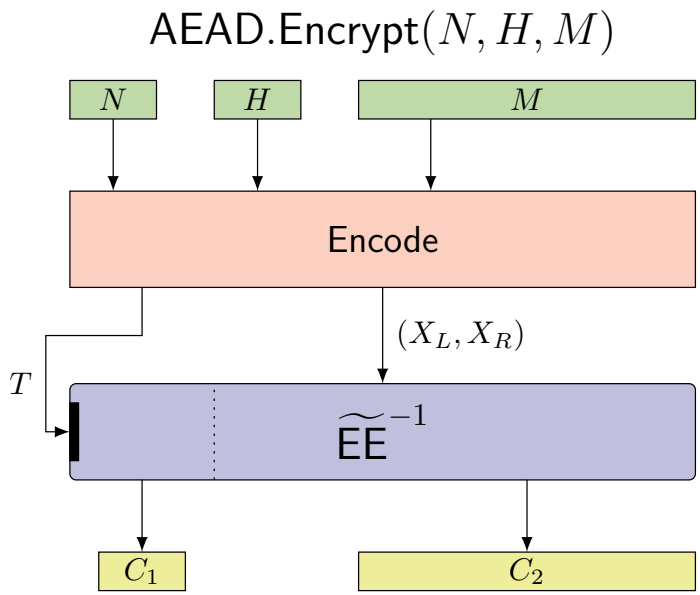


`EtE` transform yields MRAE

RPRP Applications: AEAD Schemes [DK22]

- Encode-then-Encipher paradigm [BR00] with ~~SPRP~~ RPRP!



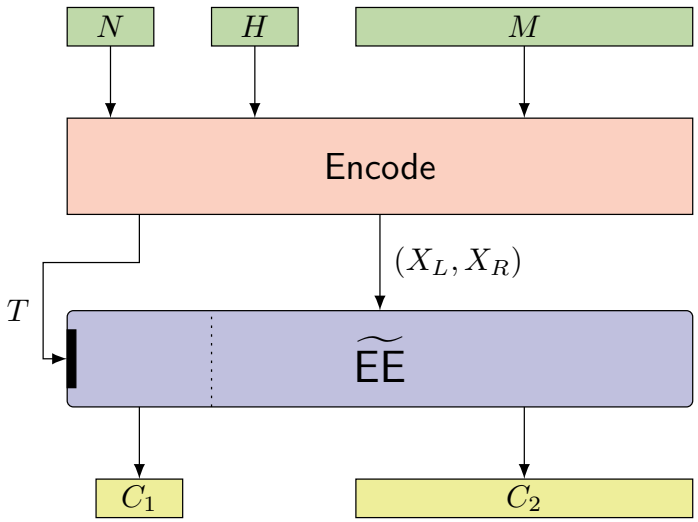


`EtE` transform yields MRAE

RPRP Applications: AEAD Schemes [DK22]

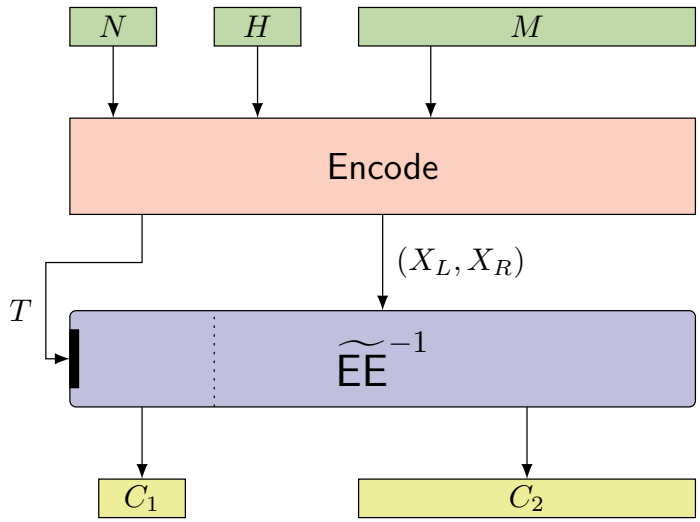
- Encode-then-Encipher paradigm [BR00] with ~~SPRP~~ RPRP!

AEAD.Encrypt(N, H, M)



`EtE` transform yields MRAE

AEAD.Encrypt(N, H, M)

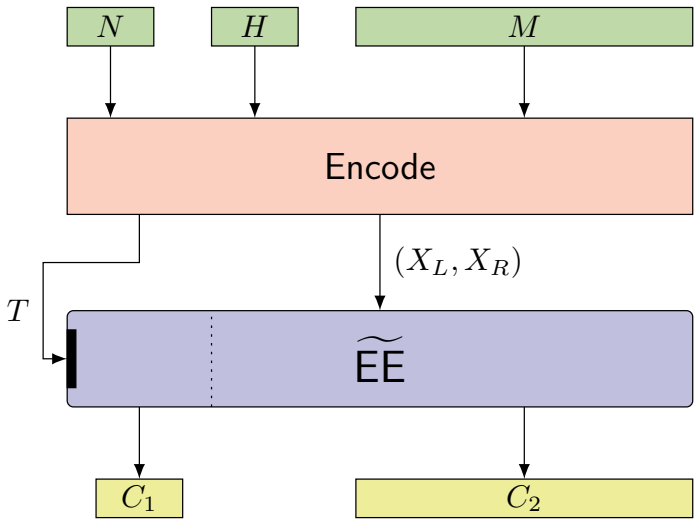


`EtD` transform yields RUPAE

RPRP Applications: AEAD Schemes [DK22]

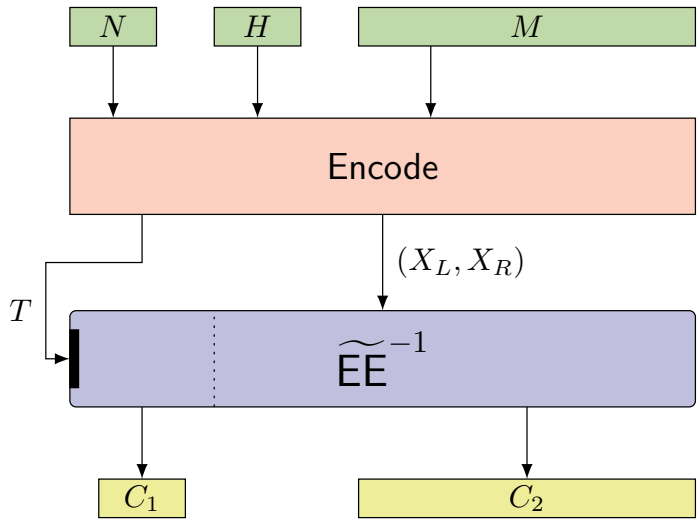
- Encode-then-Encipher paradigm [BR00] with ~~SPRP~~ RPRP!

AEAD.Encrypt(N, H, M)

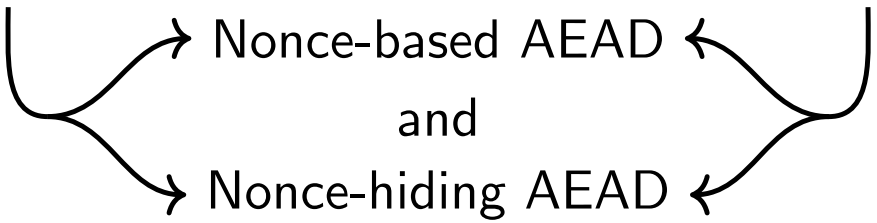


`EtE` transform yields MRAE

AEAD.Encrypt(N, H, M)



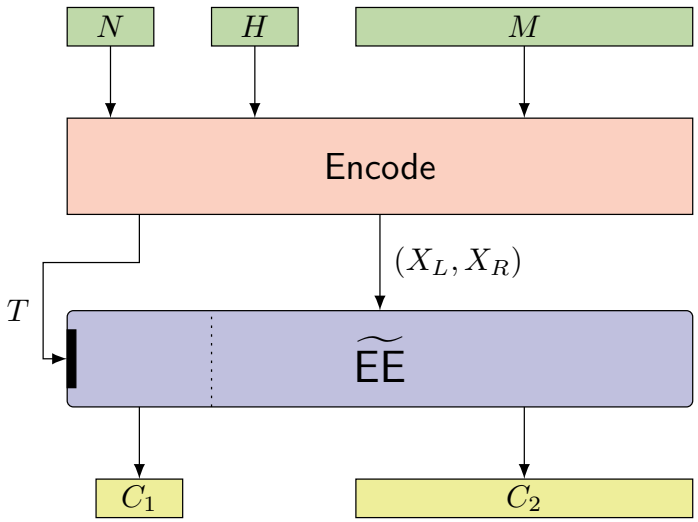
`EtD` transform yields RUPAE



RPRP Applications: AEAD Schemes [DK22]

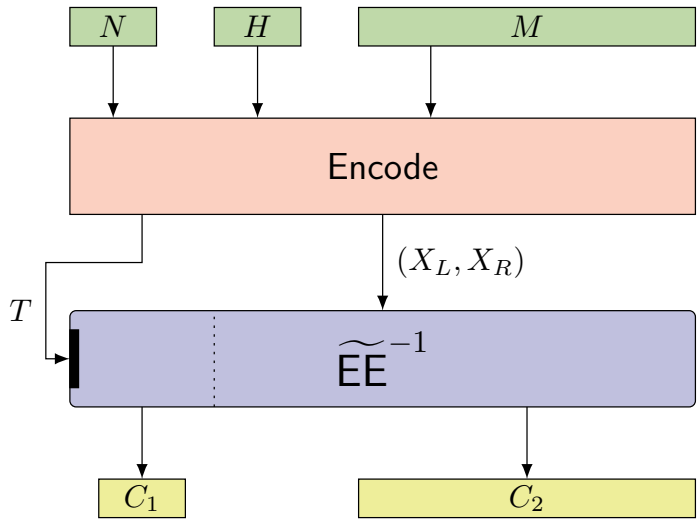
- Encode-then-Encipher paradigm [BR00] with ~~SPRP~~ RPRP!

AEAD.Encrypt(N, H, M)

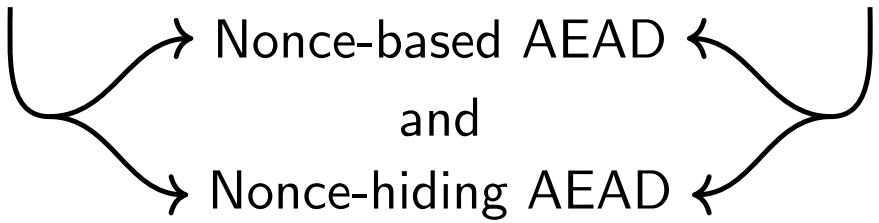


`EtE` transform yields MRAE

AEAD.Encrypt(N, H, M)

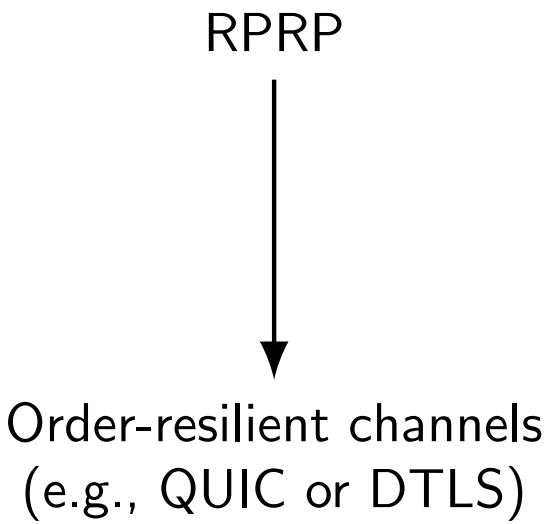


`EtD` transform yields RUPAE

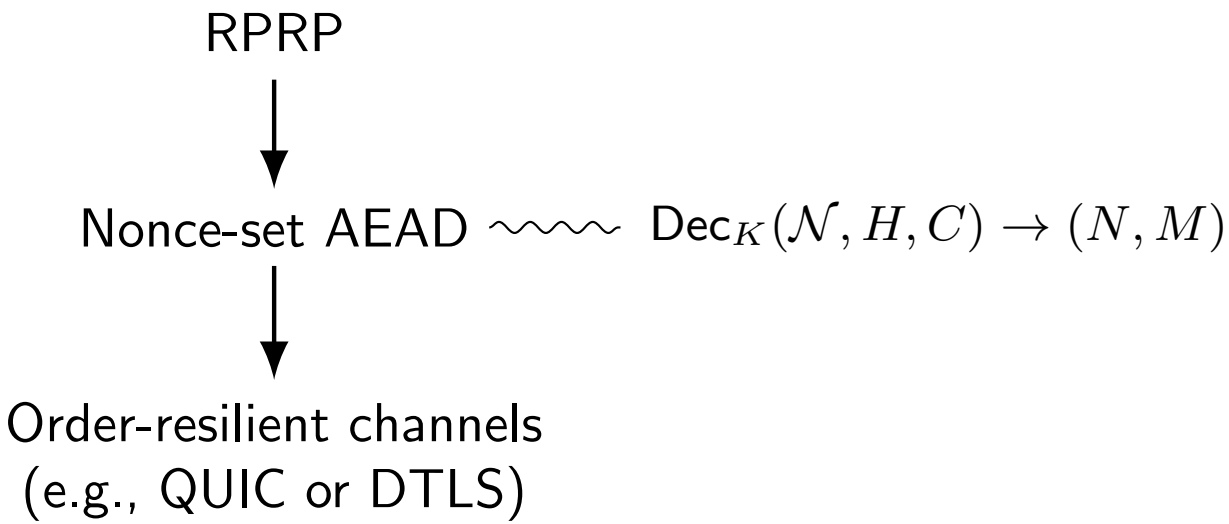


4 AEAD schemes!

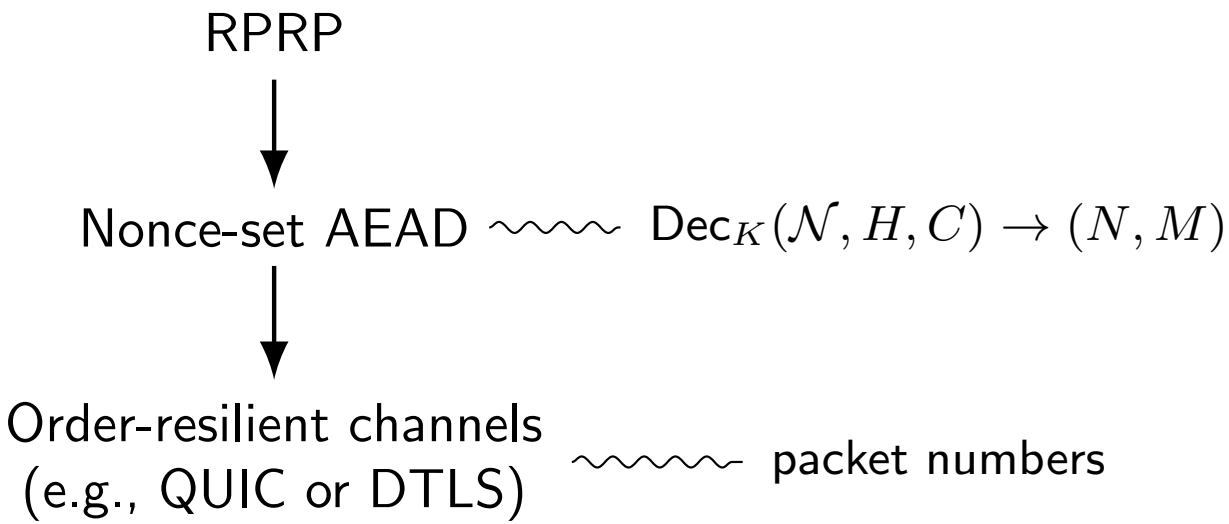
RPRP Applications: Order-Resilient Channels [DK22]



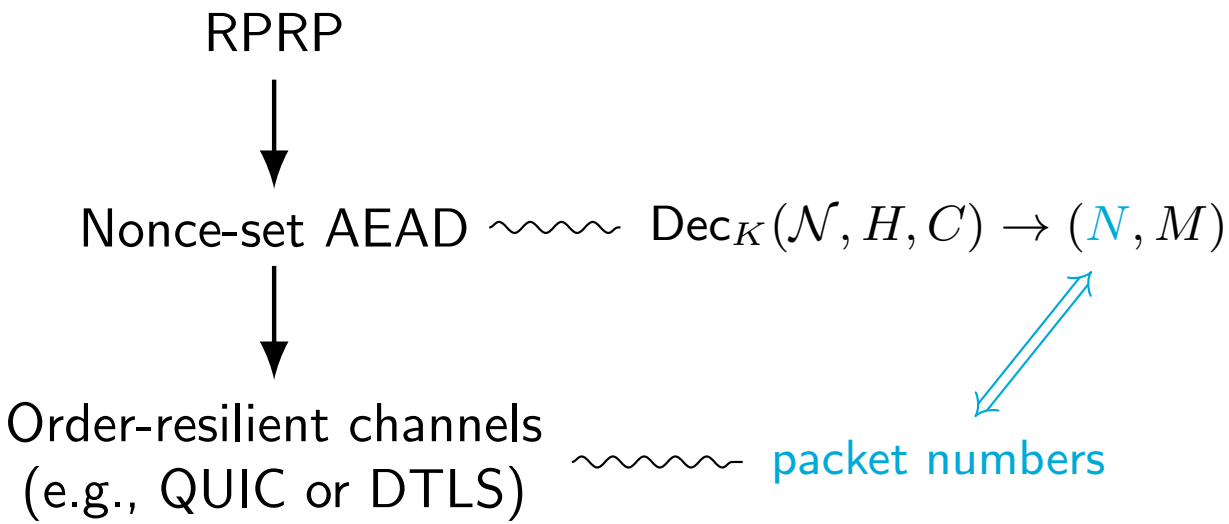
RPRP Applications: Order-Resilient Channels [DK22]



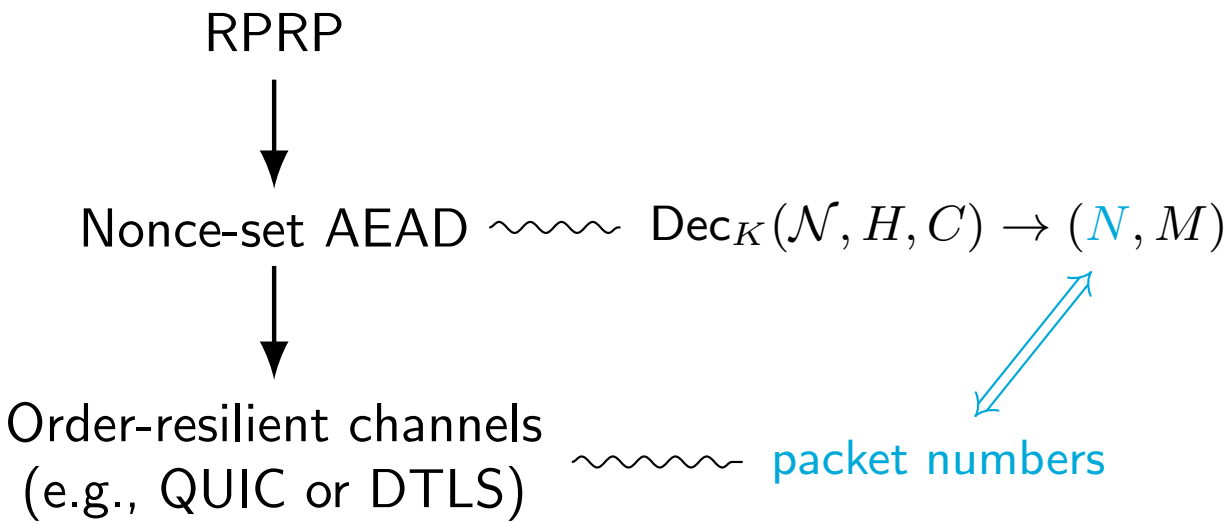
RPRP Applications: Order-Resilient Channels [DK22]



RPRP Applications: Order-Resilient Channels [DK22]



RPRP Applications: Order-Resilient Channels [DK22]



Encrypted packet numbers

✓ and

More compact ciphertext
(than, e.g., NH transforms [BNT19])

RPRP Applications: Onion Encryption [DKMMS22]

RPRP Applications: Onion Encryption [DKMMS22]

- Tagging attacks compromise privacy in Tor network [FL09].

RPRP Applications: Onion Encryption [DKMMS22]

- Tagging attacks compromise privacy in Tor network [FL09].
- Prevent these attacks by replacing each encryption layer with a wide-block cipher (i.e., SPRP).

RPRP Applications: Onion Encryption [DKMMS22]

- Tagging attacks compromise privacy in Tor network [FL09].
- Prevent these attacks by replacing each encryption layer with a wide-block cipher (i.e., SPRP).
- SPRP is an overkill $\rightarrow$ RPRP wide-block cipher is sufficient!

RPRP Applications: Onion Encryption [DKMMS22]

- Tagging attacks compromise privacy in Tor network [FL09].
- Prevent these attacks by replacing each encryption layer with a wide-block cipher (i.e., SPRP).
- SPRP is an overkill $\rightarrow$ RPRP wide-block cipher is sufficient!
- In addition, RPRP-based approach offers:

RPRP Applications: Onion Encryption [DKMMS22]

- Tagging attacks compromise privacy in Tor network [FL09].
- Prevent these attacks by replacing each encryption layer with a wide-block cipher (i.e., SPRP).
- SPRP is an overkill $\rightarrow$ RPRP wide-block cipher is sufficient!
- In addition, RPRP-based approach offers:

✓ Forward security

RPRP Applications: Onion Encryption [DKMMS22]

- Tagging attacks compromise privacy in Tor network [FL09].
- Prevent these attacks by replacing each encryption layer with a wide-block cipher (i.e., SPRP).
- SPRP is an overkill $\rightarrow$ RPRP wide-block cipher is sufficient!
- In addition, RPRP-based approach offers:

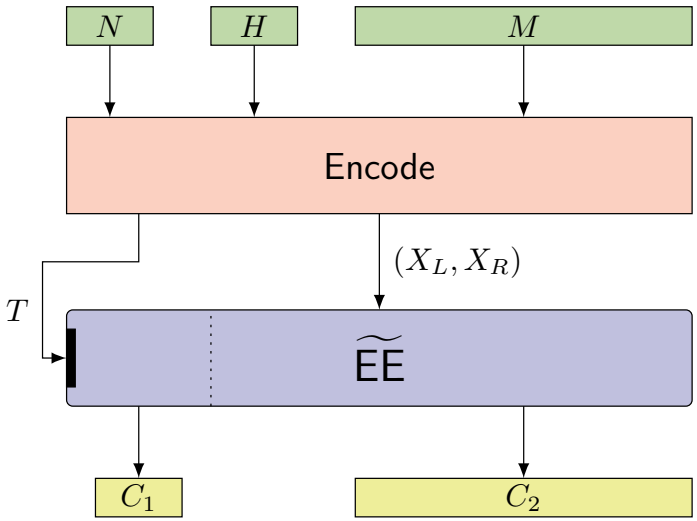
✓ Forward security

✓ Performance competitive to the Tor's current onion encryption

Revisiting the Rugged Pseudorandom Permutation Definition

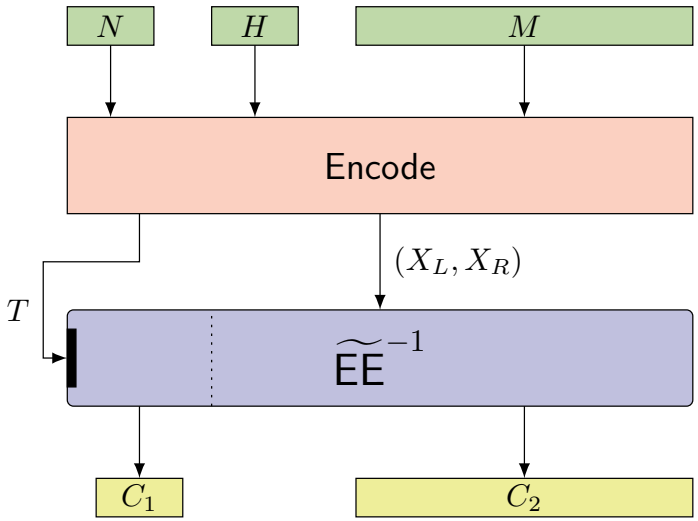
Revisiting the RPRP Definition

AEAD.Encrypt(N, H, M)



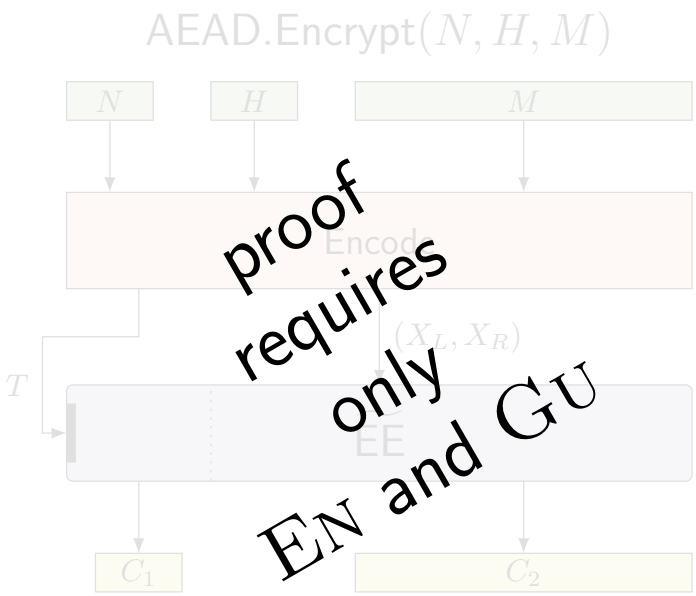
EtE transform yields MRAE

AEAD.Encrypt(N, H, M)

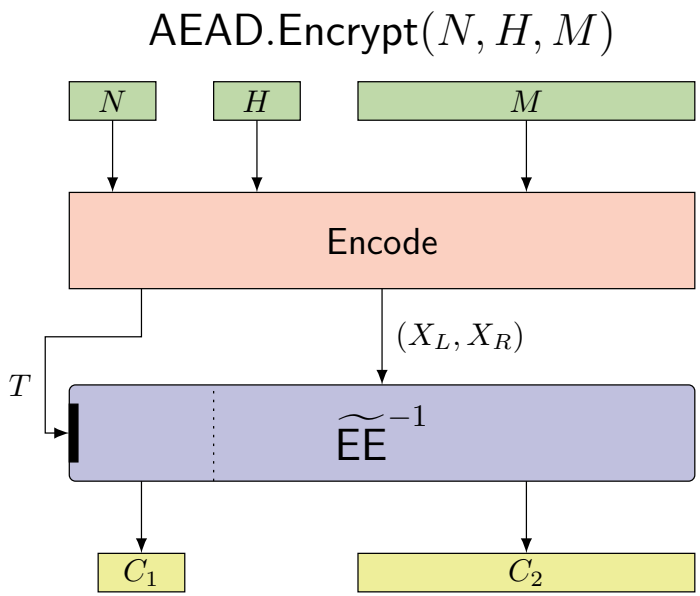


EtD transform yields RUPAE

Revisiting the RPRP Definition



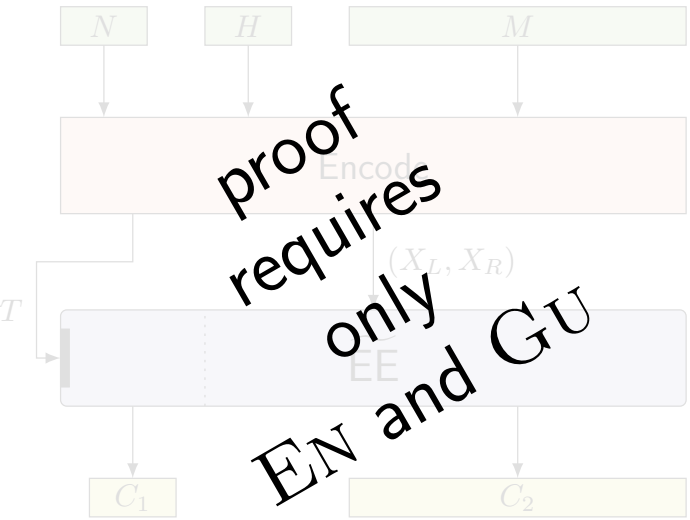
EtE transform yields MRAE



EtD transform yields RUPAE

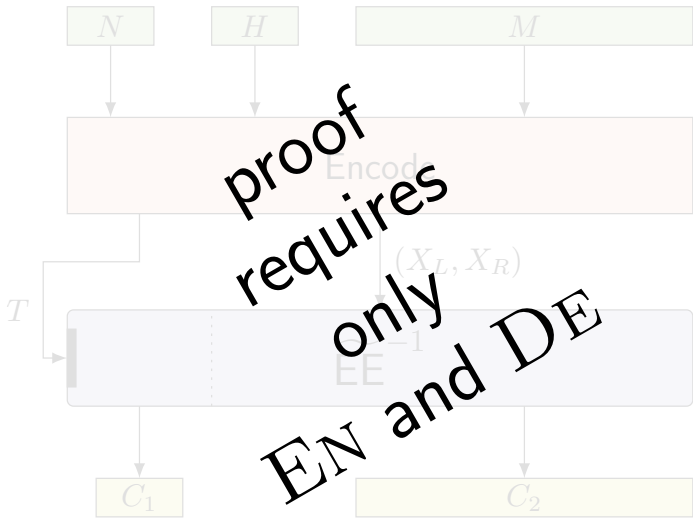
Revisiting the RPRP Definition

AEAD.Encrypt(N, H, M)



EtE transform yields MRAE

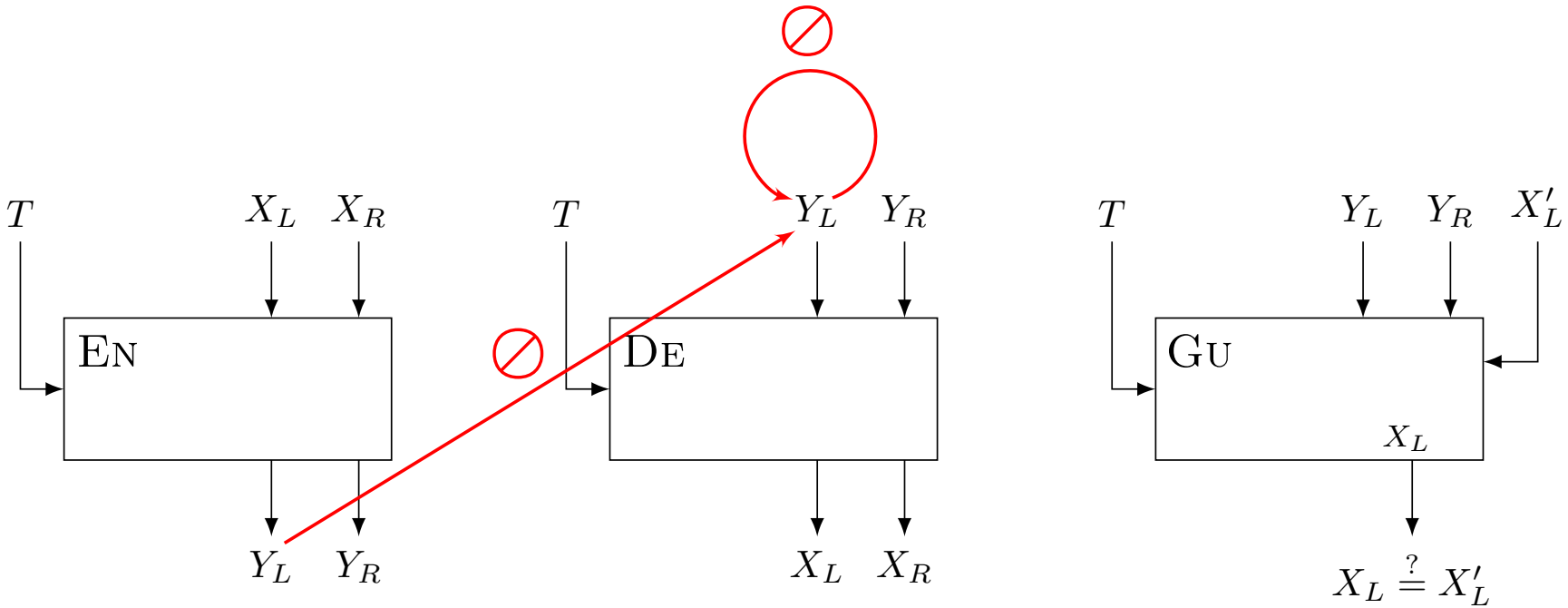
AEAD.Encrypt(N, H, M)



EtD transform yields RUPAE

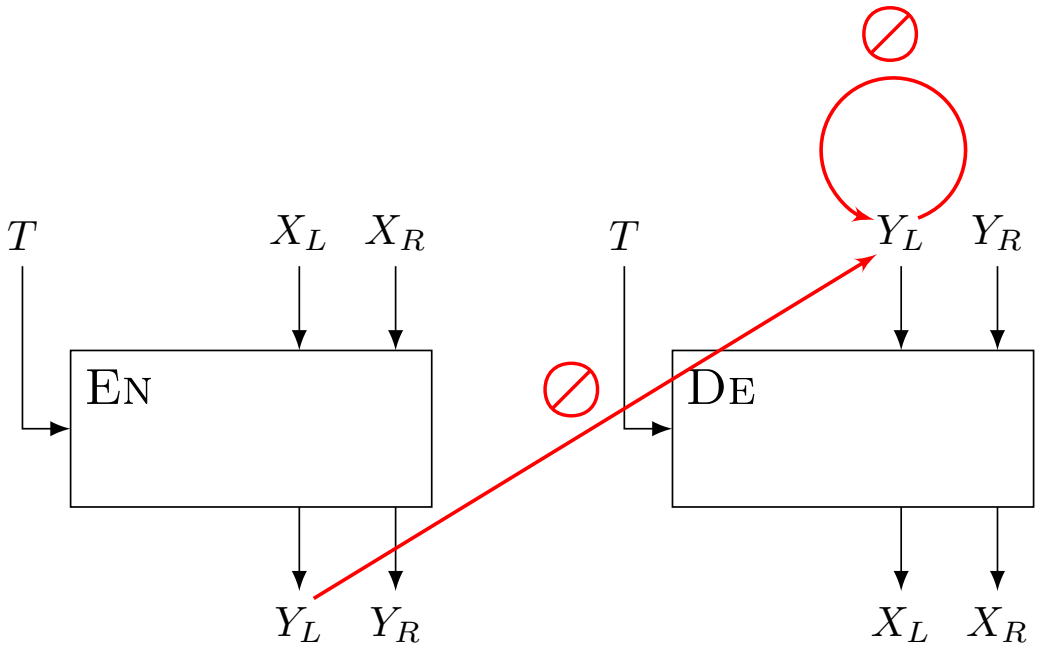
RPRP Definition Variants

RPRP



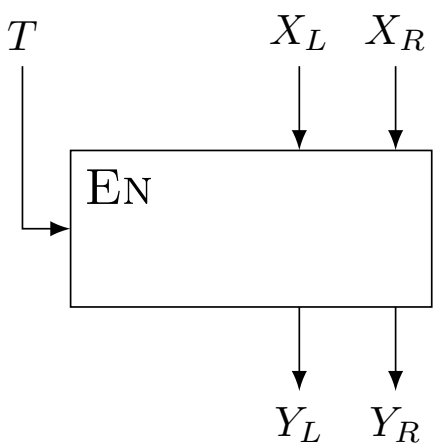
RPRP Definition Variants

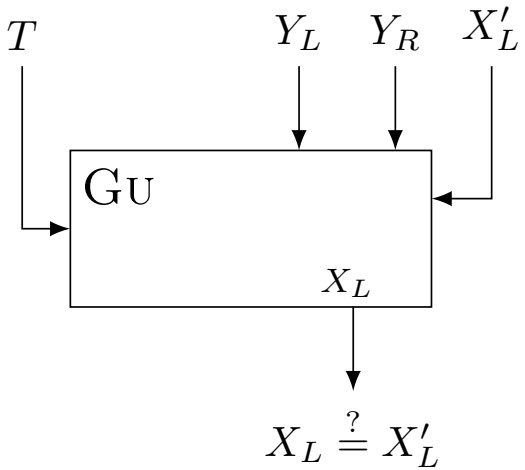
RPRPd



RPRP Definition Variants

RPRPg





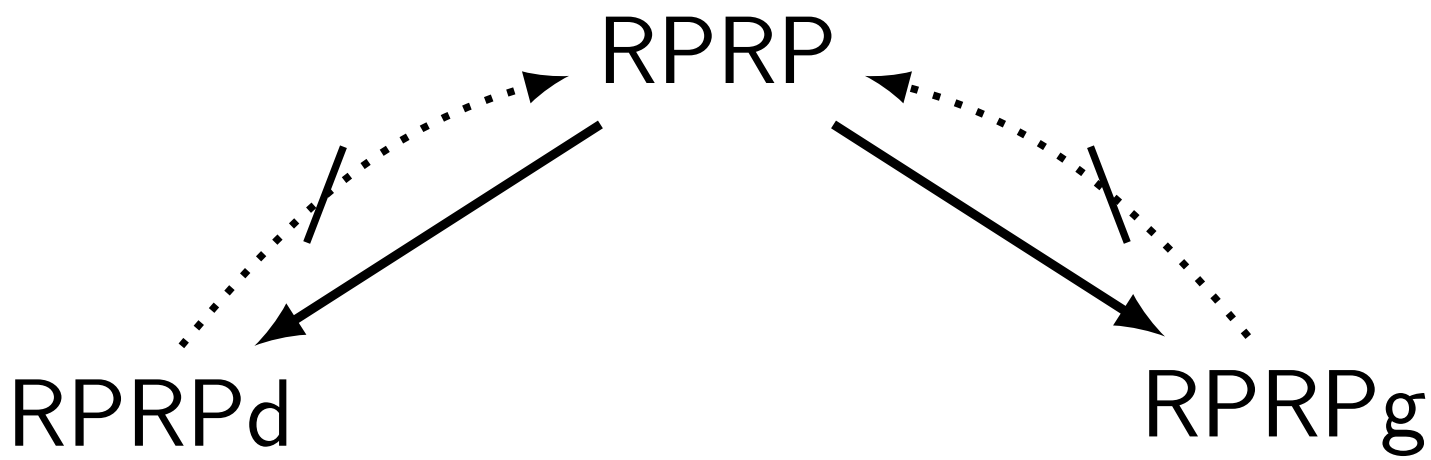
RPRP Definitions Relations

RPRP

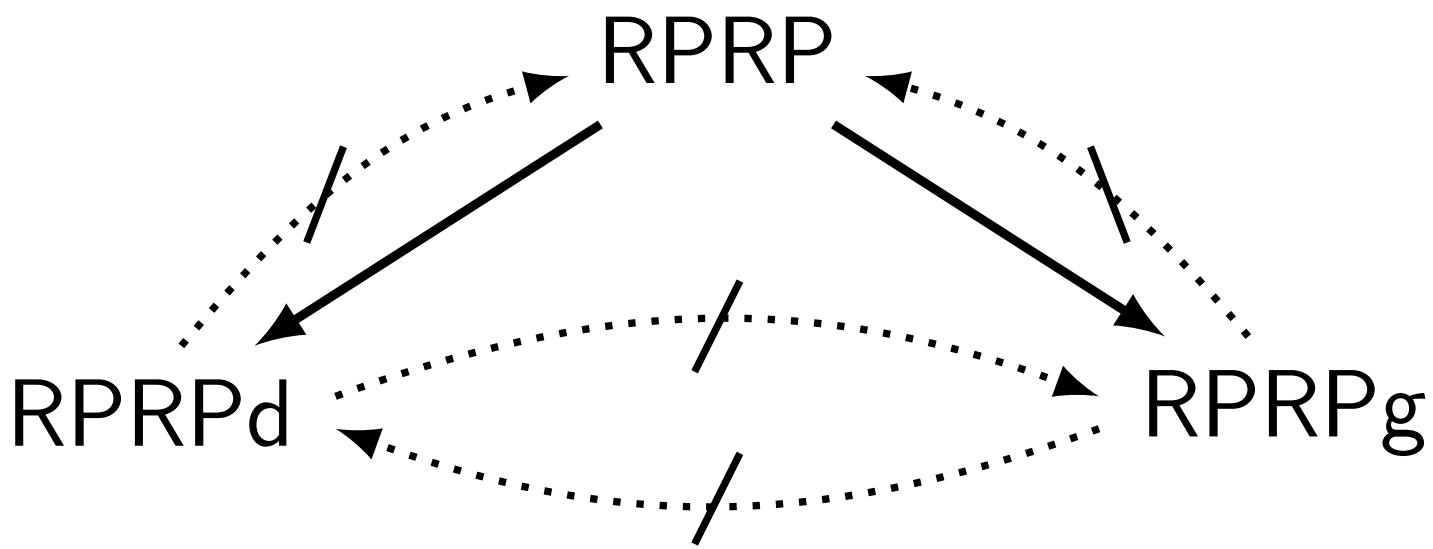
RPRPd

RPRPg

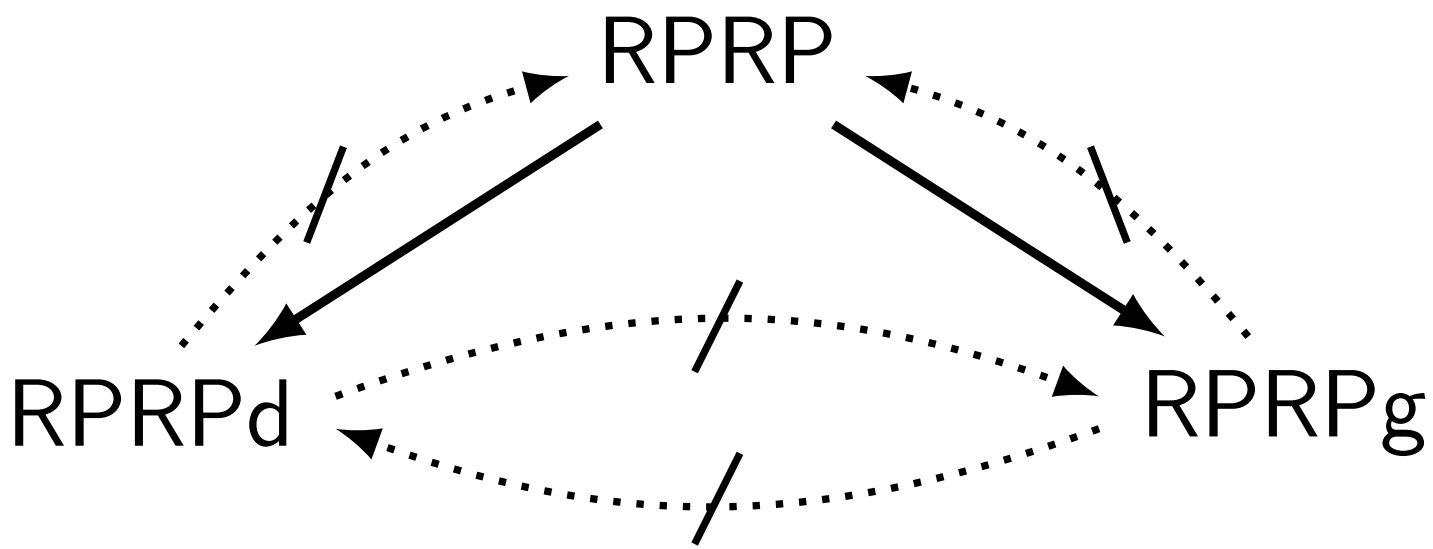
RPRP Definitions Relations



RPRP Definitions Relations

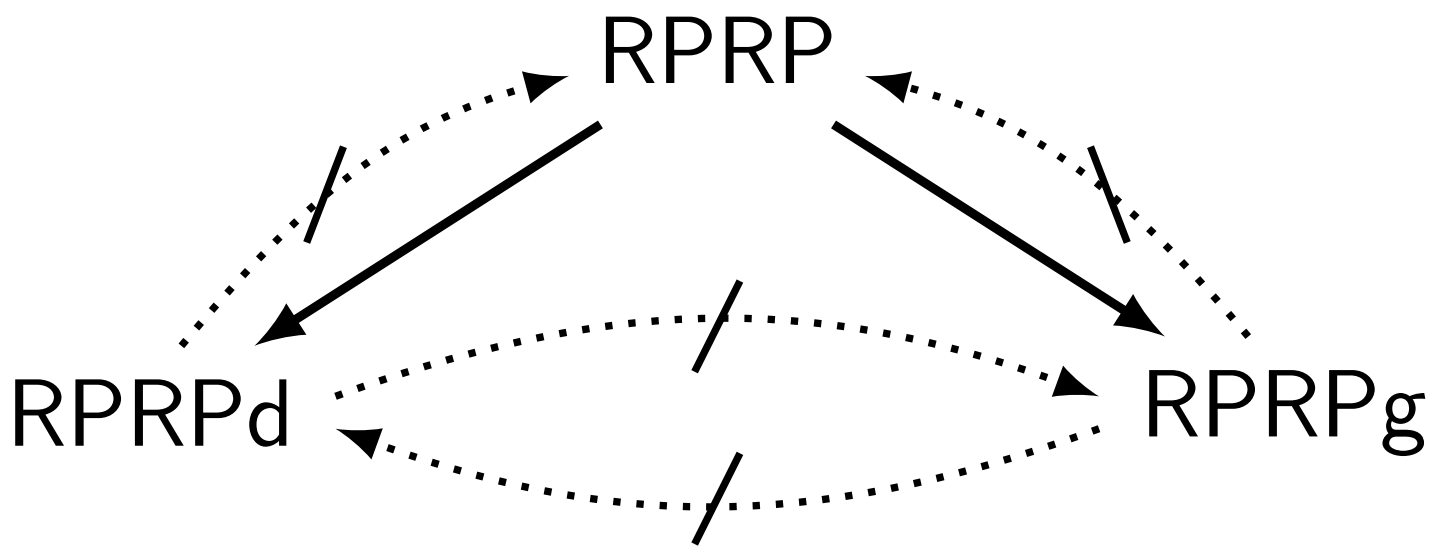


RPRP Definitions Relations



Minimized requirements
for applications!

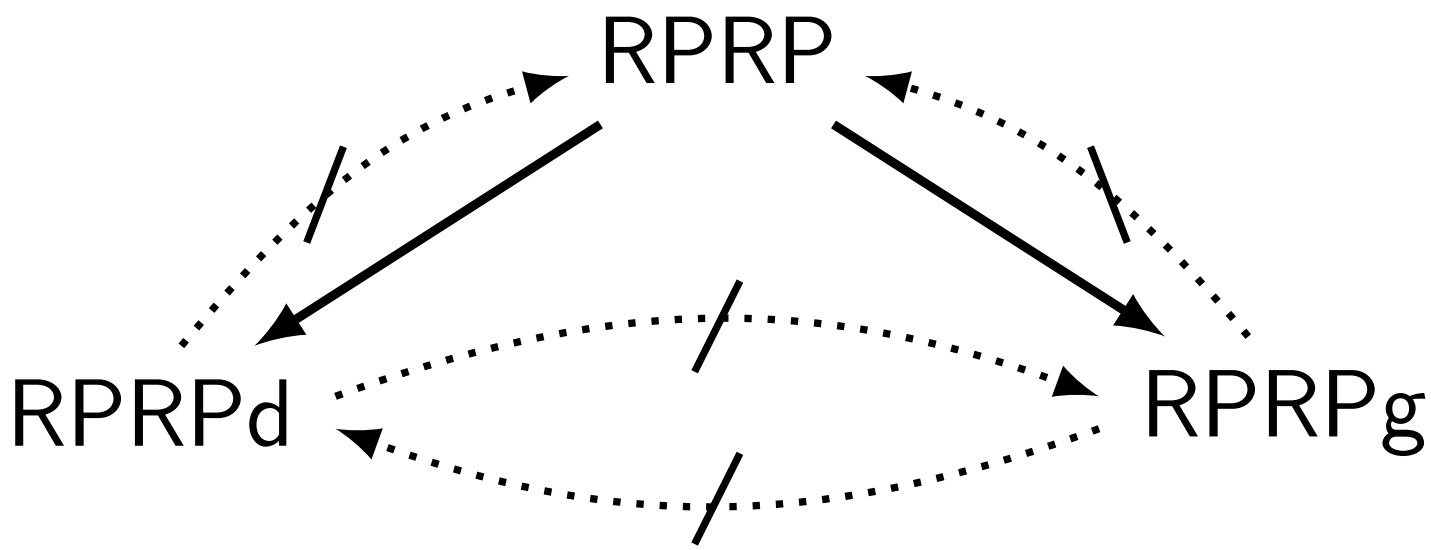
RPRP Definitions Relations



RPRP notion is still desirable!

Minimized requirements
for applications!

RPRP Definitions Relations

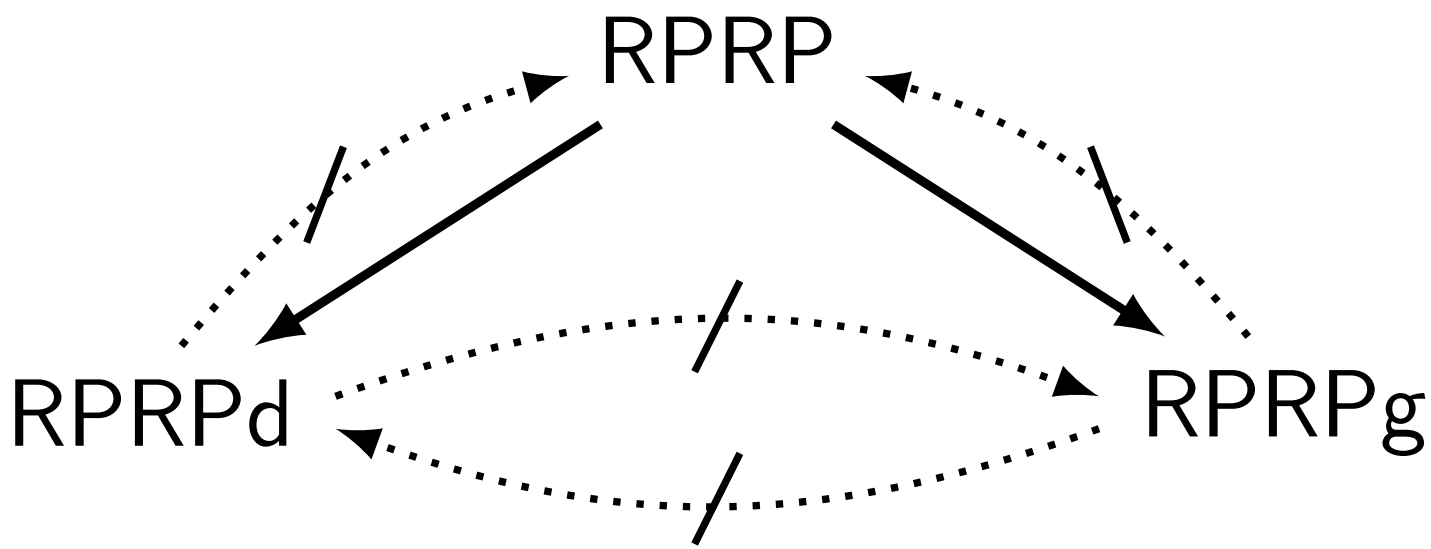


RPRP notion is still desirable!

Minimized requirements for applications!

both RPRPd and RPRPg with a single construction

RPRP Definitions Relations



Minimized requirements for applications!

RPRP notion is still desirable!

both RPRPd and RPRPg with a single construction

Onion encryption use case

Looking at the Constructions

Looking at the Constructions

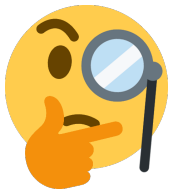
1. Reviewer 2: $\approx$ "*RPRP definition by itself looks a bit artificial.*"

Looking at the Constructions

1. Reviewer 2: $\approx$ "*RPRP definition by itself looks a bit artificial.*"
2. [DK22] gave only a single RPRP-secure construction (`UIV`).

Looking at the Constructions

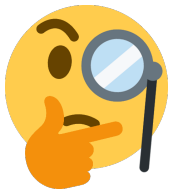
1. Reviewer 2: $\approx$ "*RPRP definition by itself looks a bit artificial.*"
2. [DK22] gave only a single RPRP-secure construction (`UIV`).



Has the notion been tailored to this single construction?

Looking at the Constructions

1. Reviewer 2: $\approx$ "*RPRP definition by itself looks a bit artificial.*"
2. [DK22] gave only a single RPRP-secure construction (`UIV`).

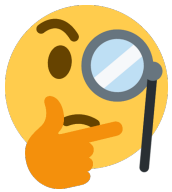


Has the notion been tailored to this single construction?

Was the RPRP abstraction worth while?

Looking at the Constructions

1. Reviewer 2: $\approx$ "*RPRP definition by itself looks a bit artificial.*"
2. [DK22] gave only a single RPRP-secure construction (`UIV`).



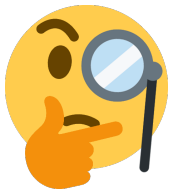
Has the notion been tailored to this single construction?

Was the RPRP abstraction worth while?

3 new constructions

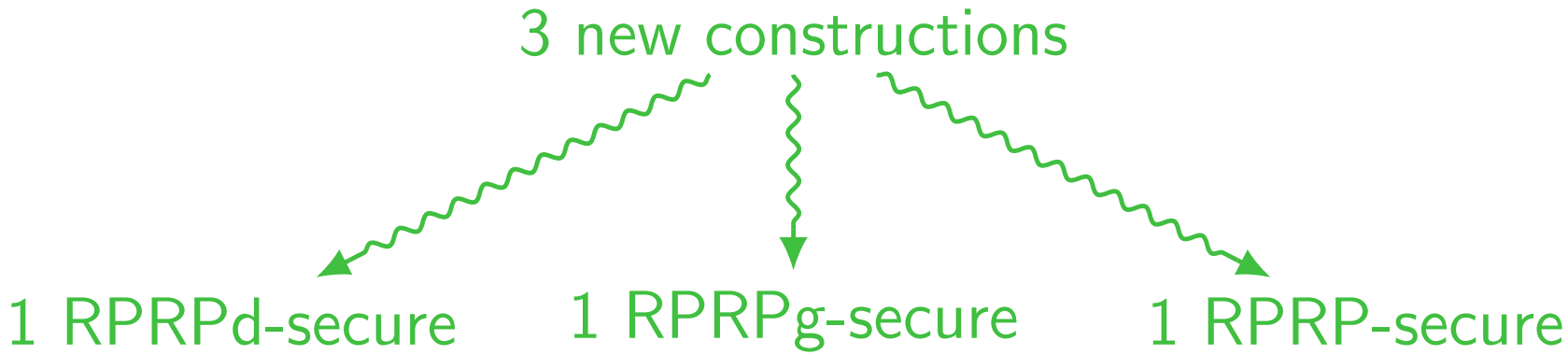
Looking at the Constructions

1. Reviewer 2: $\approx$ "*RPRP definition by itself looks a bit artificial.*"
2. [DK22] gave only a single RPRP-secure construction (`UIV`).



Has the notion been tailored to this single construction?

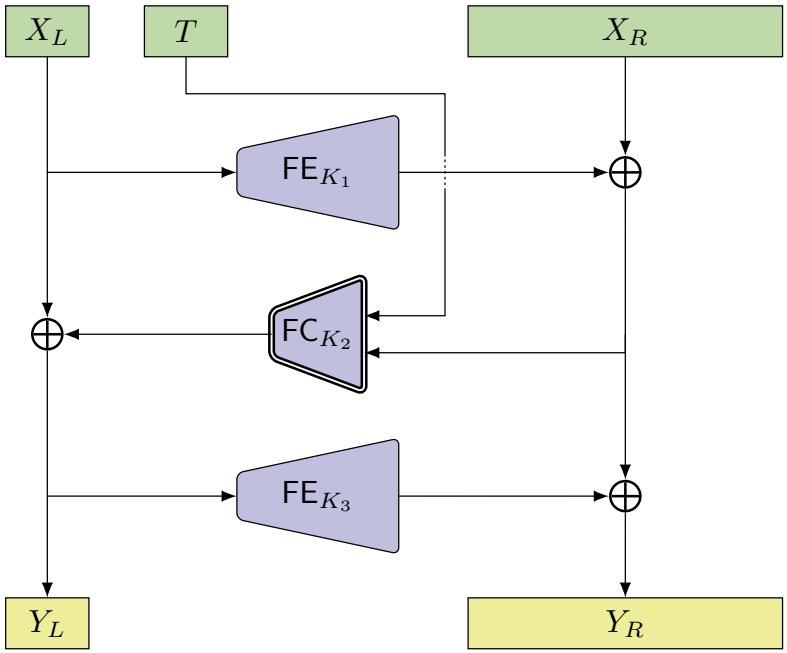
Was the RPRP abstraction worth while?



Three-Round Feistel Constructions

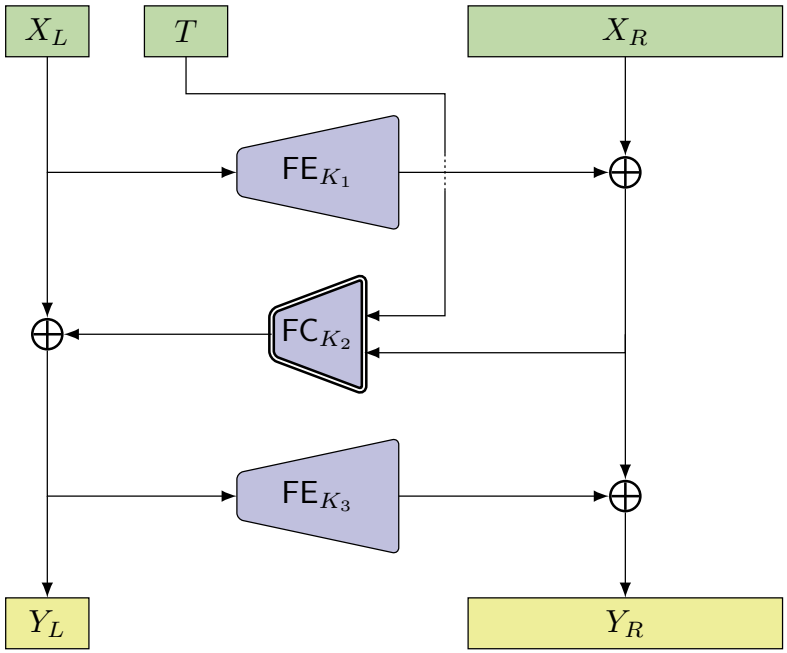
Three-Round Feistel Constructions

ECE (Expand-Compress-Expand)



Three-Round Feistel Constructions

ECE (Expand-Compress-Expand)

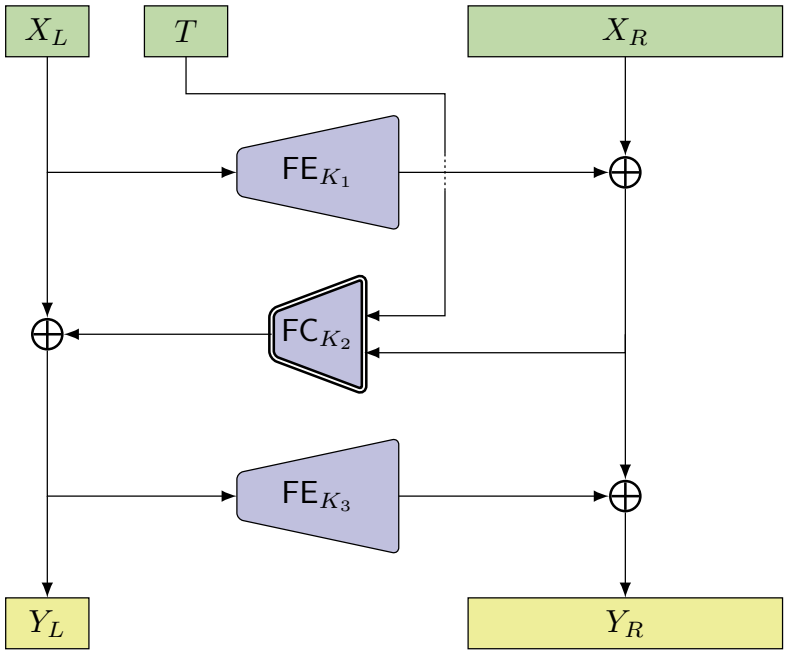


RPRPd secure

proof

Three-Round Feistel Constructions

ECE (Expand-Compress-Expand)

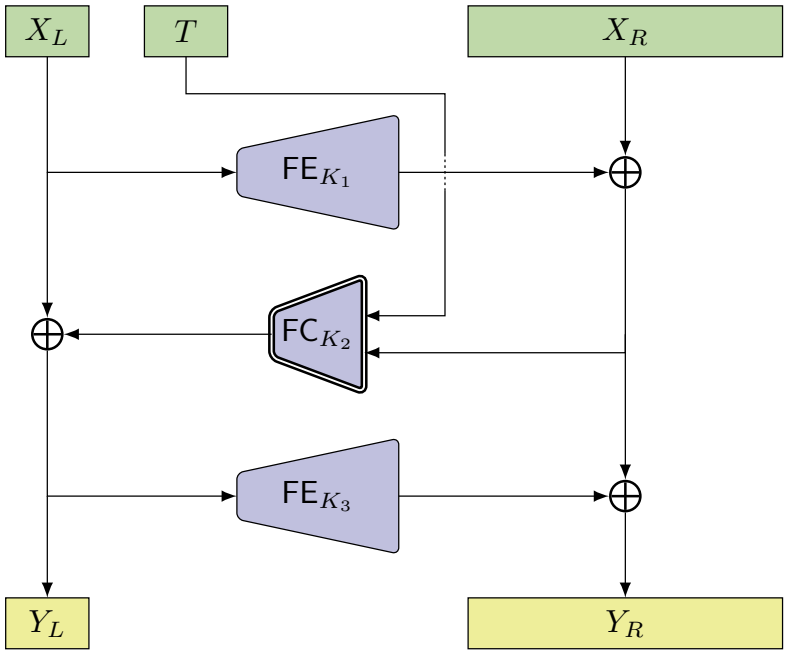


RPRPd secure
proof

~~RPRPg secure~~
attack

Three-Round Feistel Constructions

ECE (Expand-Compress-Expand)



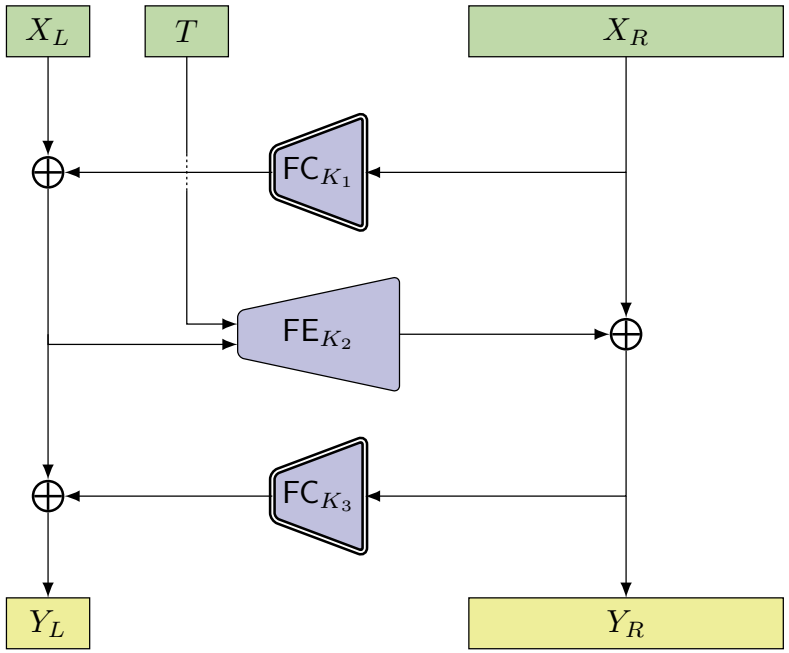
RPRPd secure

proof

~~RPRPg secure~~

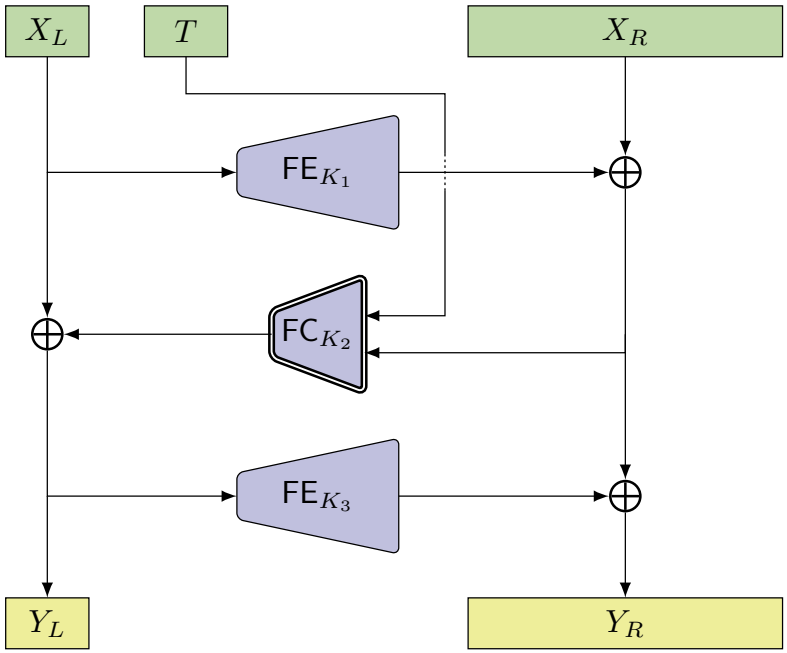
attack

CEC (Compress-Expand-Compress)



Three-Round Feistel Constructions

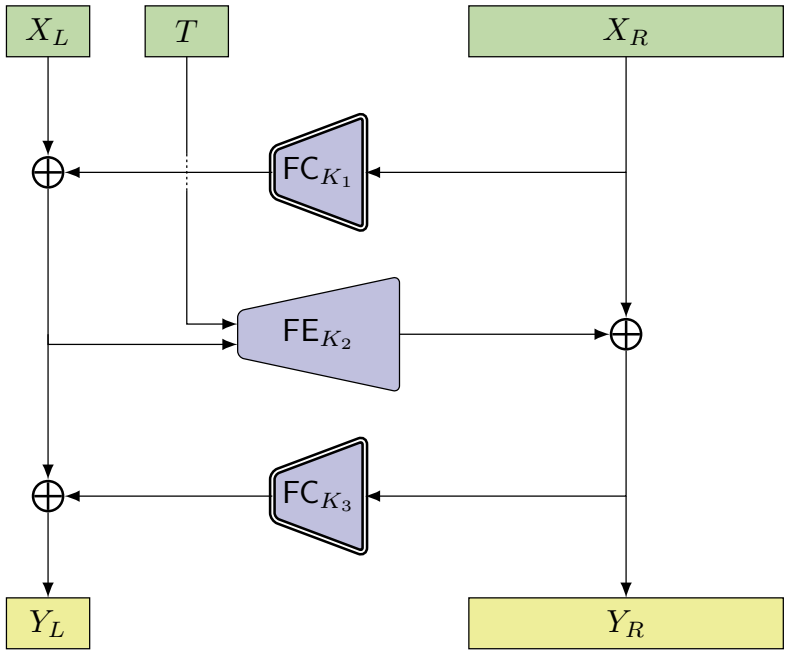
ECE (Expand-Compress-Expand)



RPRPd secure
proof

~~RPRPg secure~~
attack

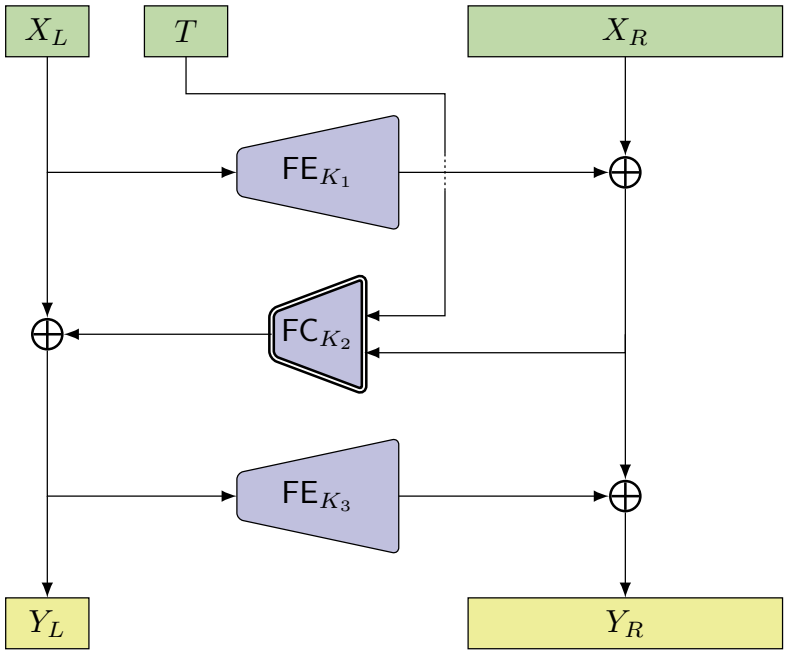
CEC (Compress-Expand-Compress)



RPRPg secure
proof

Three-Round Feistel Constructions

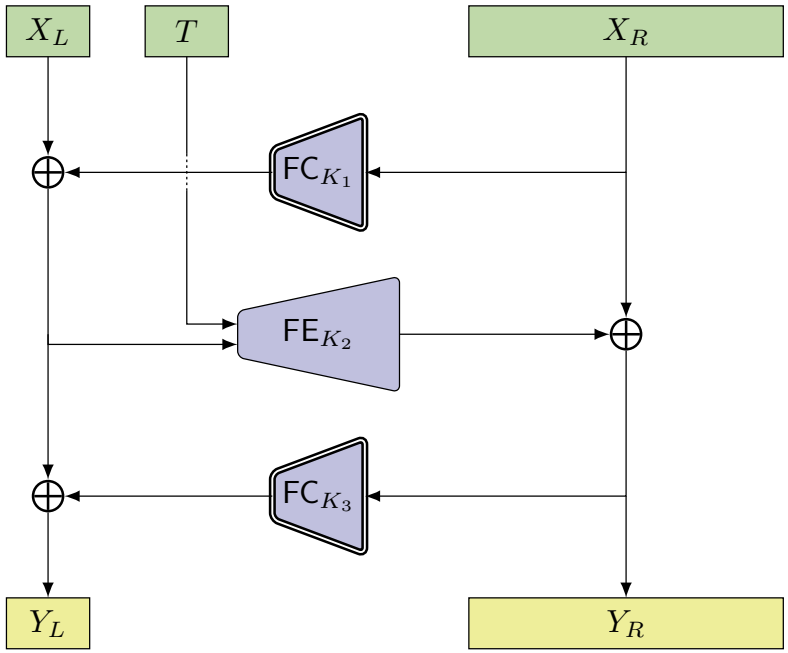
ECE (Expand-Compress-Expand)



RPRPd secure
proof

~~RPRPg secure~~
attack

CEC (Compress-Expand-Compress)

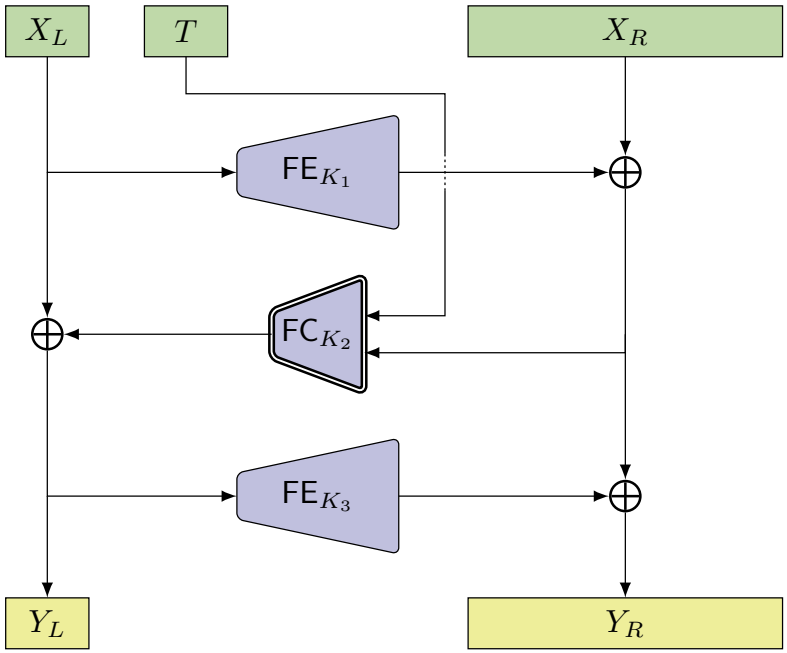


RPRPg secure
proof

~~RPRPd secure~~
attack

Three-Round Feistel Constructions

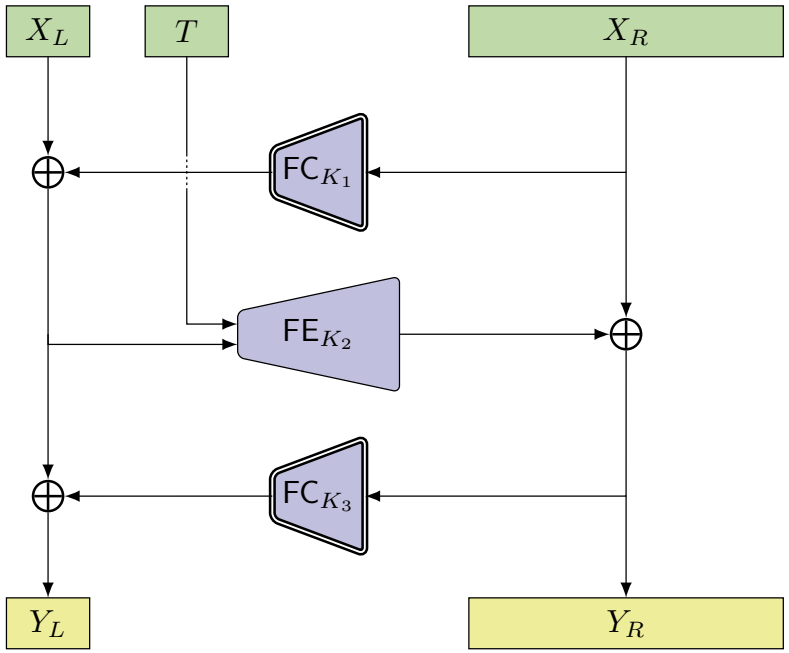
`ECE` (Expand-Compress-Expand)



RPRPd secure
proof

~~RPRPg secure~~
attack

`CEC` (Compress-Expand-Compress)



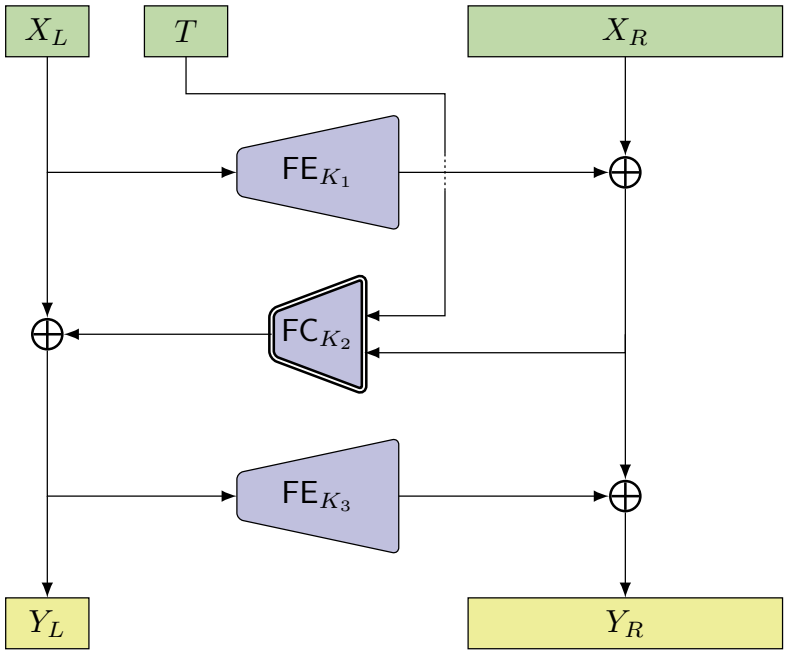
RPRPg secure
proof

~~RPRPd secure~~
attack

- Three-round Feistel achieves (cipher-based) notion **stronger** than PRP [LR88].

Three-Round Feistel Constructions

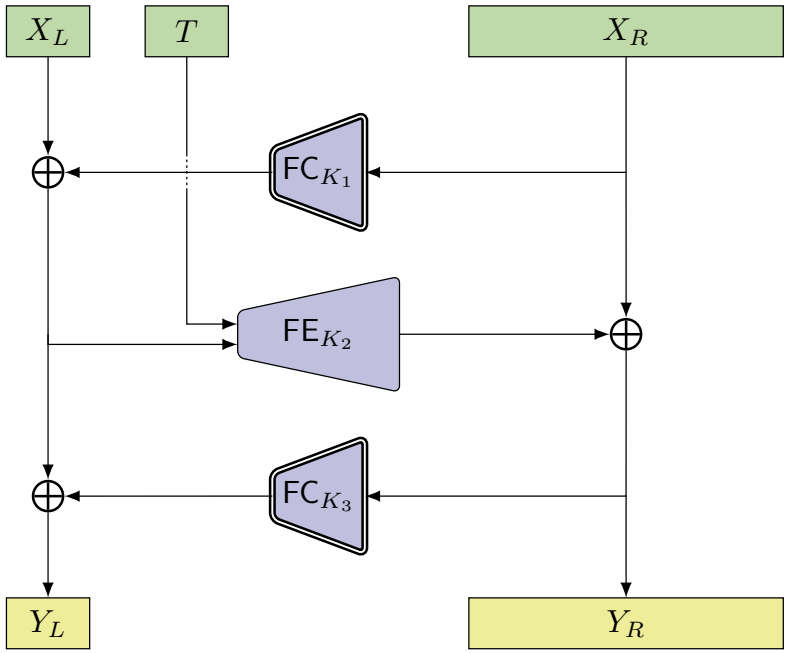
ECE (Expand-Compress-Expand)



RPRPd secure
proof

~~RPRPg secure~~
attack

CEC (Compress-Expand-Compress)



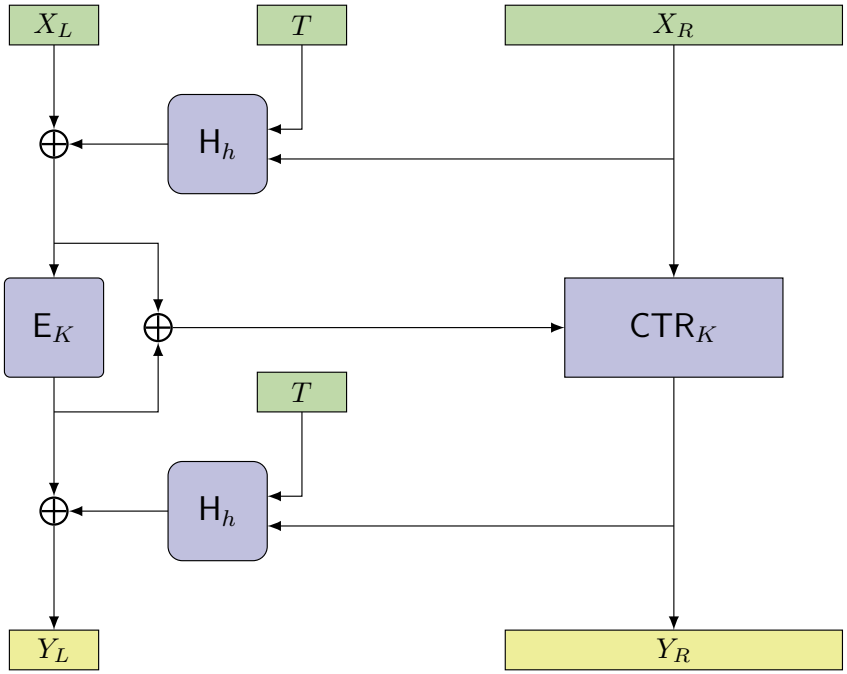
RPRPg secure
proof

~~RPRPd secure~~
attack

- Three-round Feistel achieves (cipher-based) notion **stronger** than PRP [LR88].
- Fast instantiations possible with deck functions [BDHAK22].

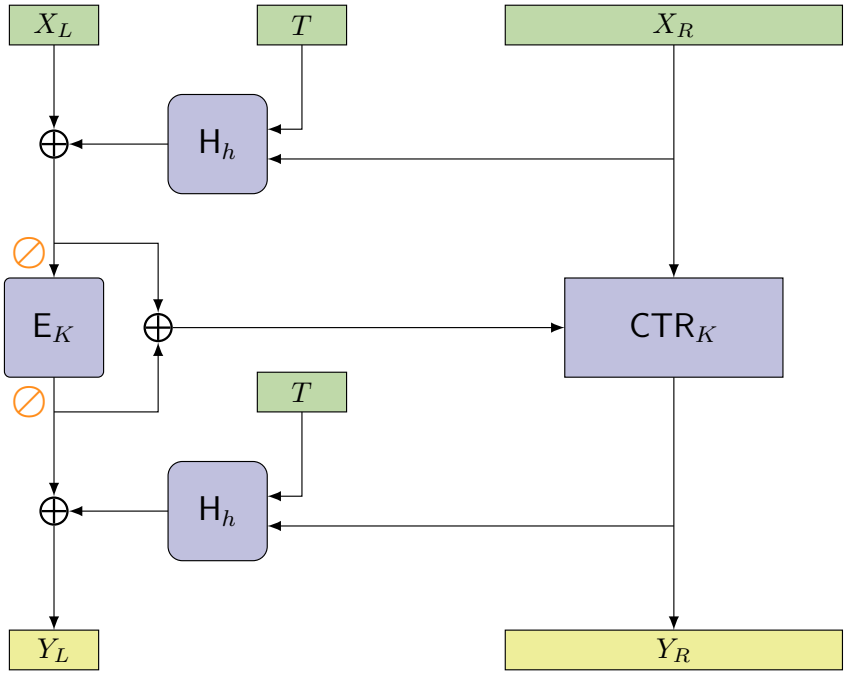
HEC Construction

HEC Construction



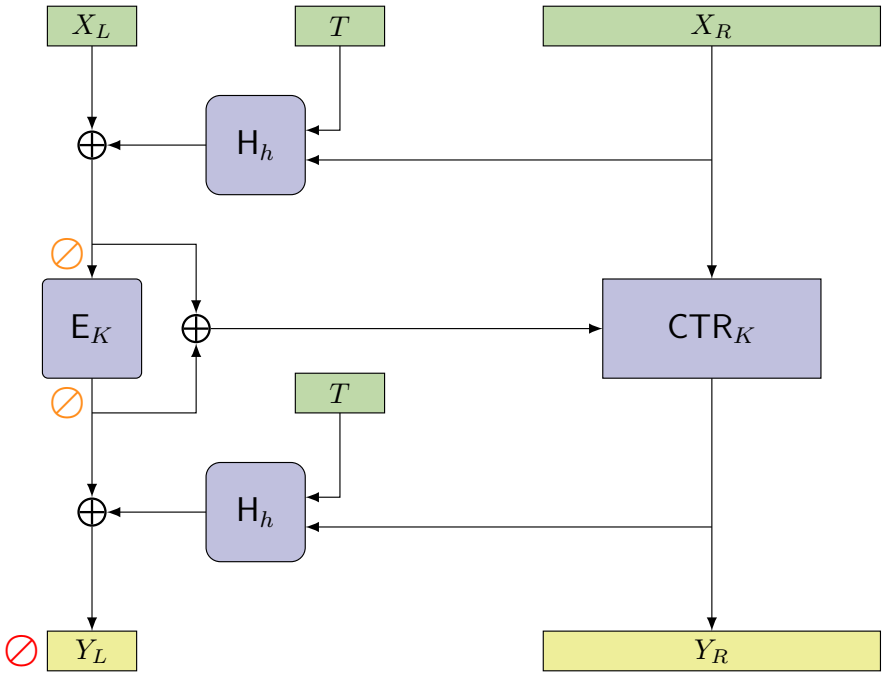
- HCTR construction [WFW05] as a base.

HEC Construction



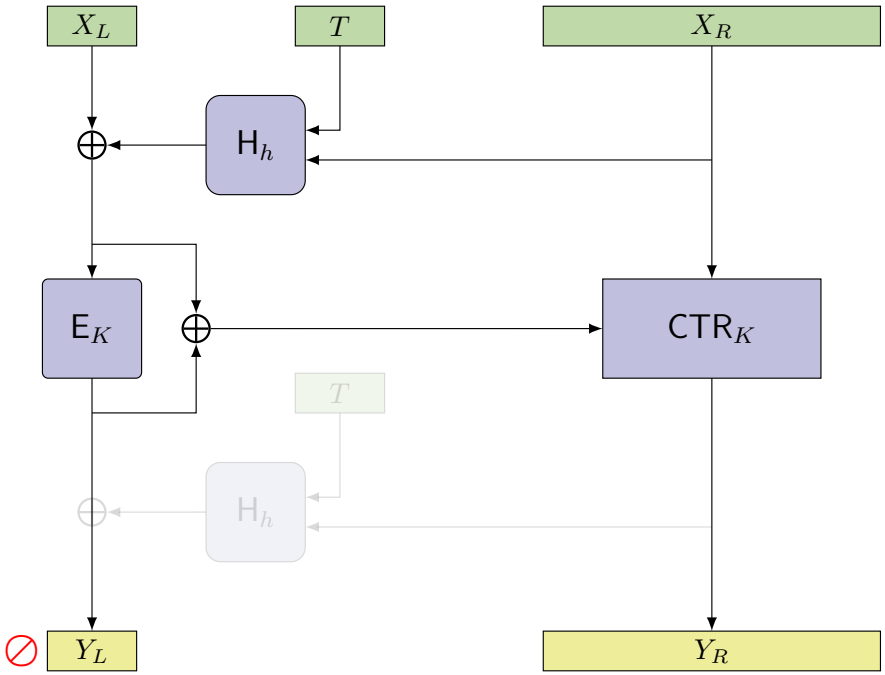
- HCTR construction [WFW05] as a base.

HEC Construction



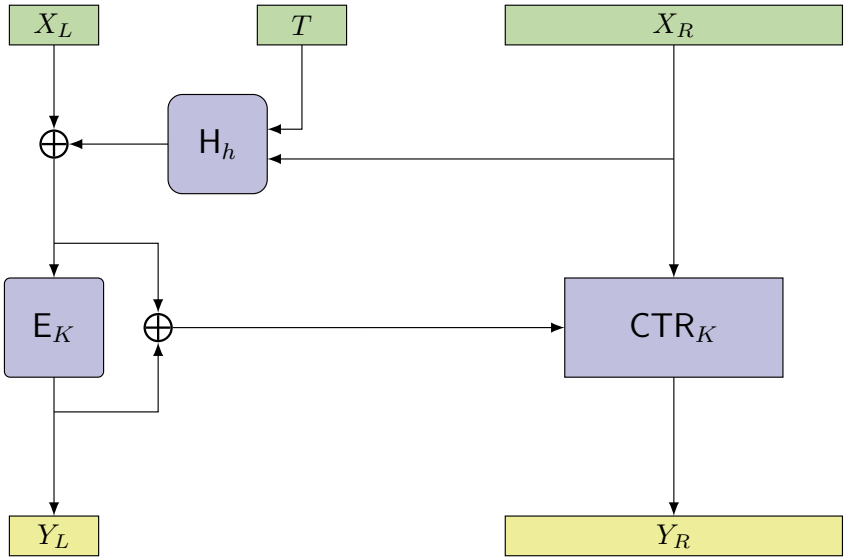
- HCTR construction [WFW05] as a base.

HEC Construction



- HCTR construction [WFW05] as a base.
- HEC:
  1. remove lower layer

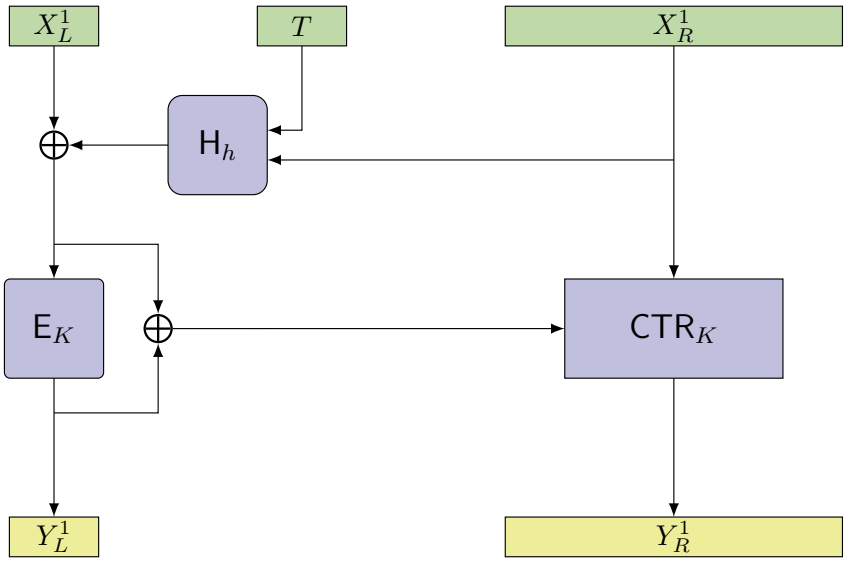
HEC Construction



~~RPRP secure~~

- HCTR construction [WFW05] as a base.
- HEC:
  1. remove lower layer

HEC Construction

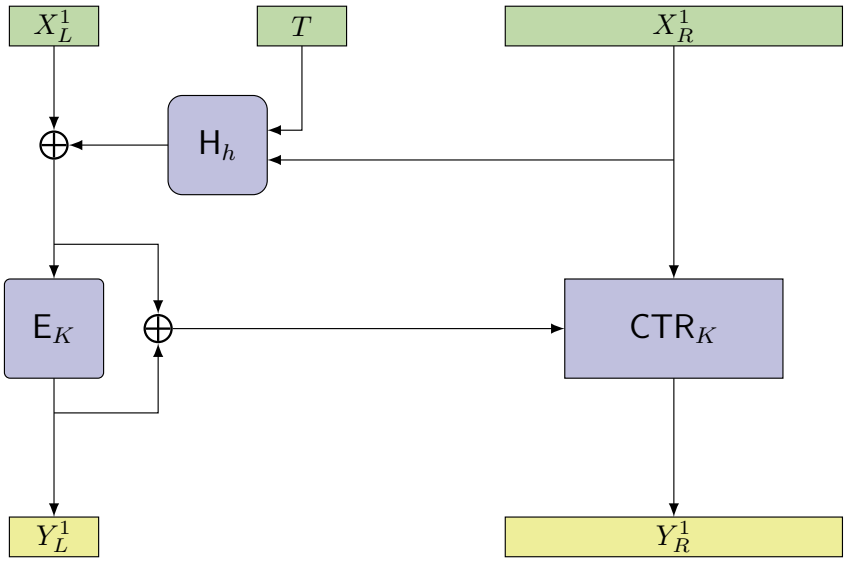


~~RPRP secure~~

1. $(Y_L^1, Y_R^1) \leftarrow \text{EN}(T, X_L^1, X_R^1)$

- HCTR construction [WFW05] as a base.
- HEC:
  1. remove lower layer

HEC Construction



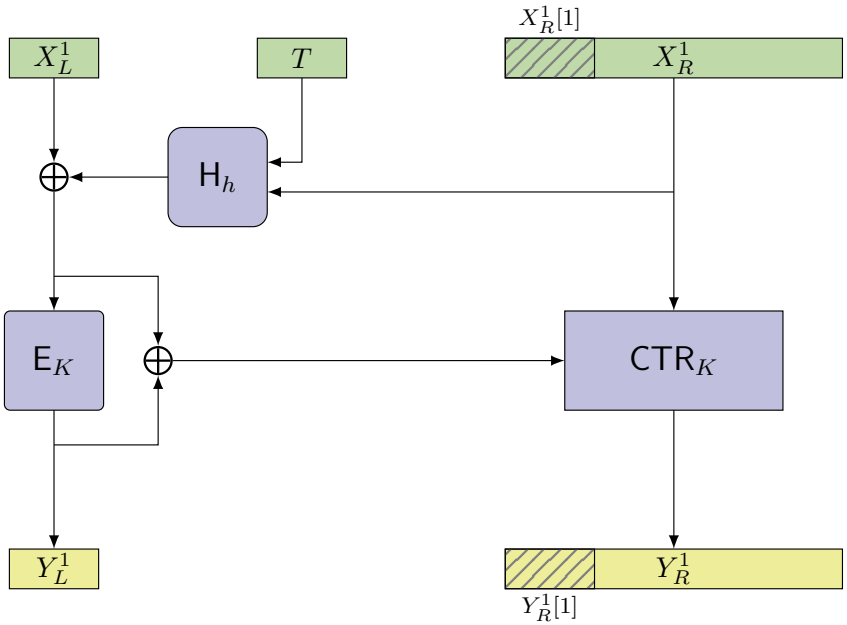
~~RPRP secure~~

1. $(Y_L^1, Y_R^1) \leftarrow \text{EN}(T, X_L^1, X_R^1)$

$\rightarrow H_h(T, X_R^1) = X_L^1 \oplus Y_L^1 \oplus IV^1$

- HCTR construction [WFW05] as a base.
- HEC:
  1. remove lower layer

HEC Construction



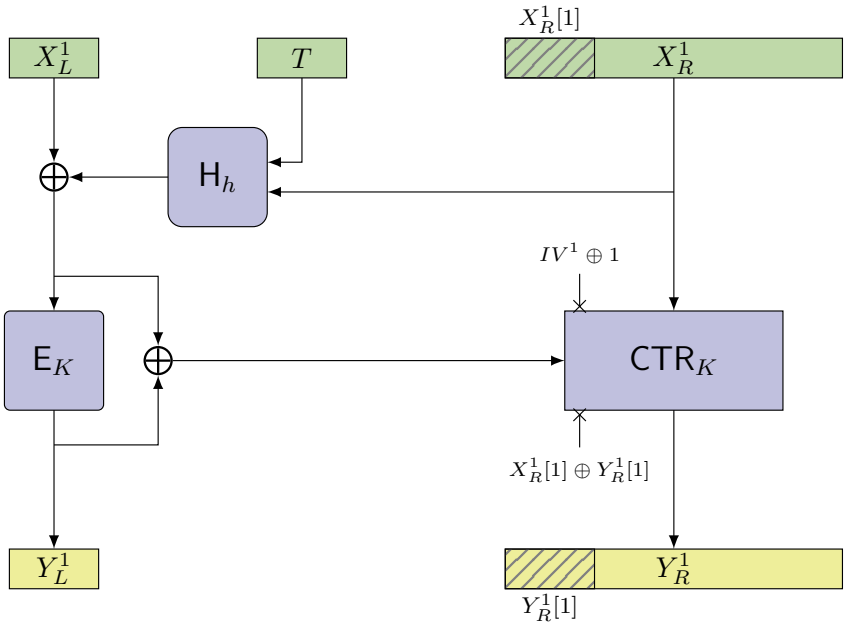
~~RPRP secure~~

1. $(Y_L^1, Y_R^1) \leftarrow \text{EN}(T, X_L^1, X_R^1)$

$\rightarrow H_h(T, X_R^1) = X_L^1 \oplus Y_L^1 \oplus IV^1$

- HCTR construction [WFW05] as a base.
- HEC:
  1. remove lower layer

HEC Construction



~~RPRP secure~~

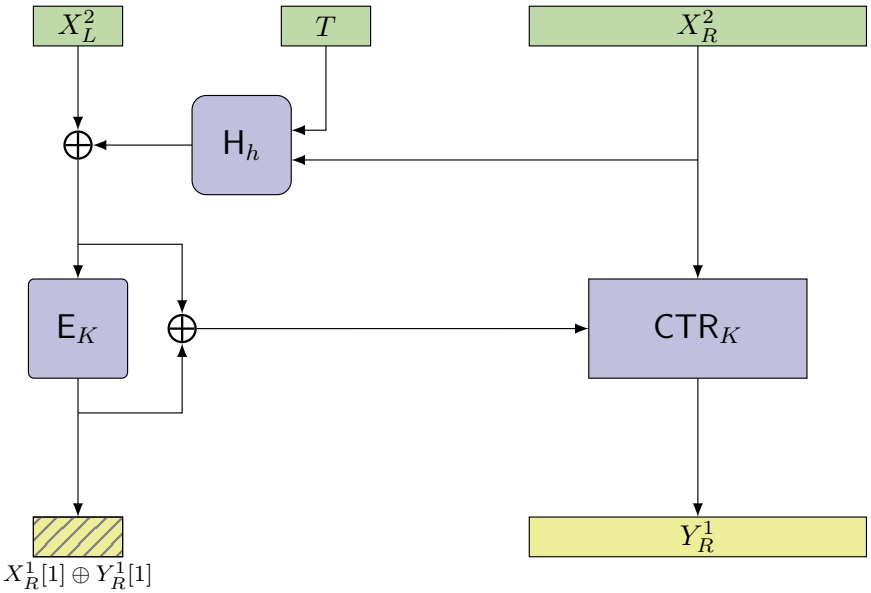
1. $(Y_L^1, Y_R^1) \leftarrow \mathrm{EN}(T, X_L^1, X_R^1)$

$\rightarrow H_h(T, X_R^1) = X_L^1 \oplus Y_L^1 \oplus IV^1$

(and $E_K(IV^1 \oplus 1) = X_R^1[1] \oplus Y_R^1[1]$)

- HCTR construction [WFW05] as a base.
- HEC:
  1. remove lower layer

HEC Construction



~~RPRP secure~~

1. $(Y_L^1, Y_R^1) \leftarrow \text{EN}(T, X_L^1, X_R^1)$

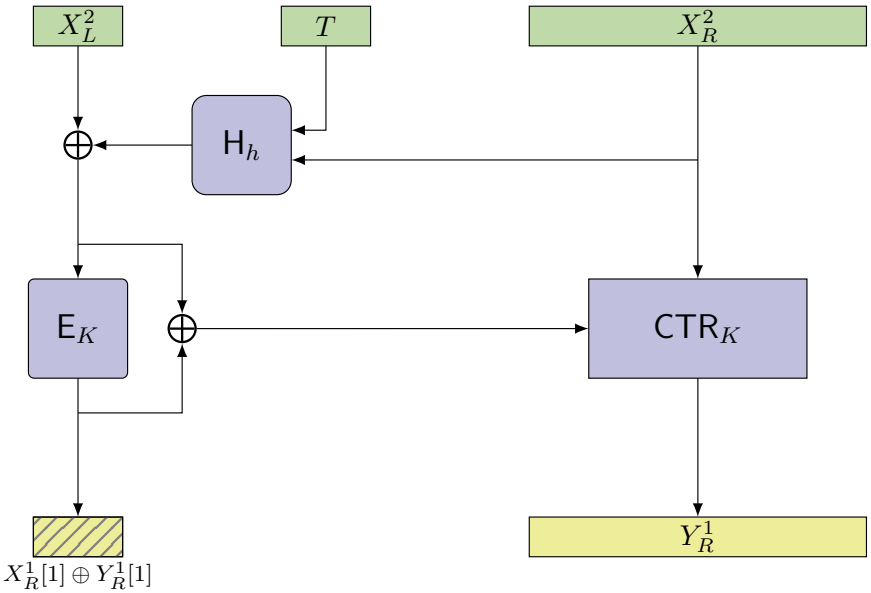
$\rightarrow H_h(T, X_R^1) = X_L^1 \oplus Y_L^1 \oplus IV^1$

(and $E_K(IV^1 \oplus 1) = X_R^1[1] \oplus Y_R^1[1]$)

2. $(X_L^2, X_R^2) \leftarrow \text{DE}(T, Y_R^1[1] \oplus X_R^1[1], Y_R^1)$

- HCTR construction [WFW05] as a base.
- HEC:
  1. remove lower layer

HEC Construction

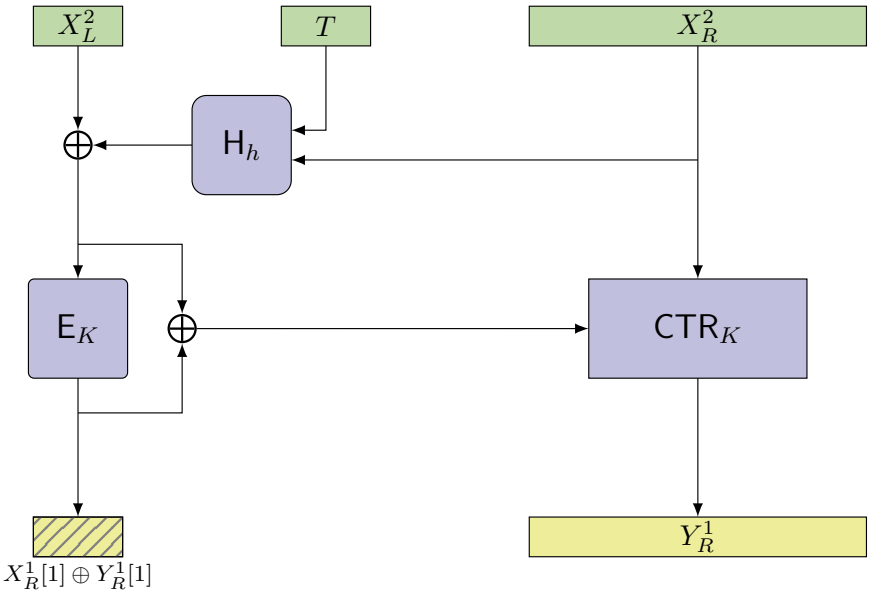


~~RPRP secure~~

1. $(Y_L^1, Y_R^1) \leftarrow \text{EN}(T, X_L^1, X_R^1)$
   $\rightarrow H_h(T, X_R^1) = X_L^1 \oplus Y_L^1 \oplus IV^1$
   (and $E_K(IV^1 \oplus 1) = X_R^1[1] \oplus Y_R^1[1]$)
2. $(X_L^2, X_R^2) \leftarrow \text{DE}(T, Y_R^1[1] \oplus X_R^1[1], Y_R^1)$
   $\rightarrow H_h(T, X_R^2) = X_L^2 \oplus IV^1 \oplus 1$

- HCTR construction [WFW05] as a base.
- HEC:
  1. remove lower layer

HEC Construction



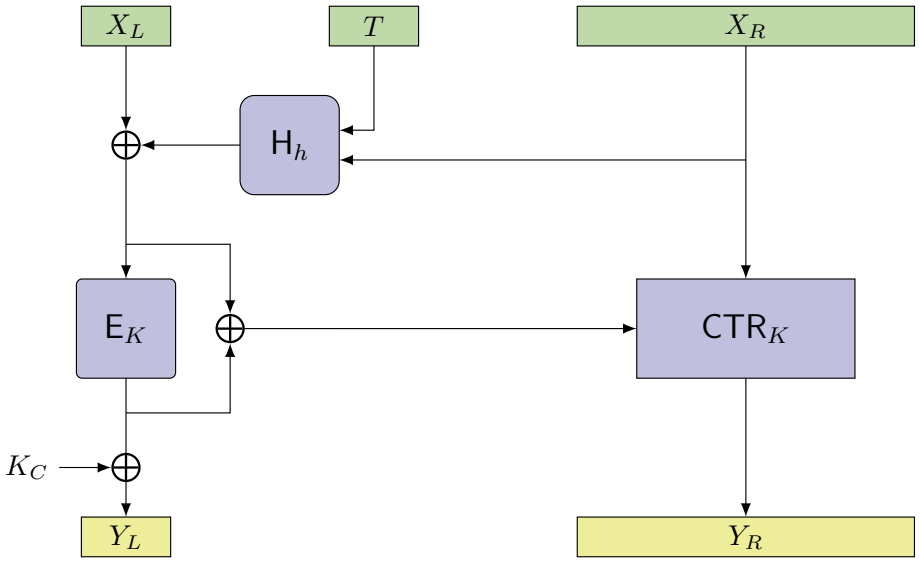
~~RPRP secure~~

1. $(Y_L^1, Y_R^1) \leftarrow \text{EN}(T, X_L^1, X_R^1)$
   $\rightarrow H_h(T, X_R^1) = X_L^1 \oplus Y_L^1 \oplus IV^1$
   (and $E_K(IV^1 \oplus 1) = X_R^1[1] \oplus Y_R^1[1]$)
2. $(X_L^2, X_R^2) \leftarrow \text{DE}(T, Y_R^1[1] \oplus X_R^1[1], Y_R^1)$
   $\rightarrow H_h(T, X_R^2) = X_L^2 \oplus IV^1 \oplus 1$
3. Calculate $H_h(T, X_R^1) \oplus H_h(T, X_R^2)$ and extract key h.

- HCTR construction [WFW05] as a base.
- HEC:
  1. remove lower layer

HEC Construction

HEC (Hash-Encipher-Counter)

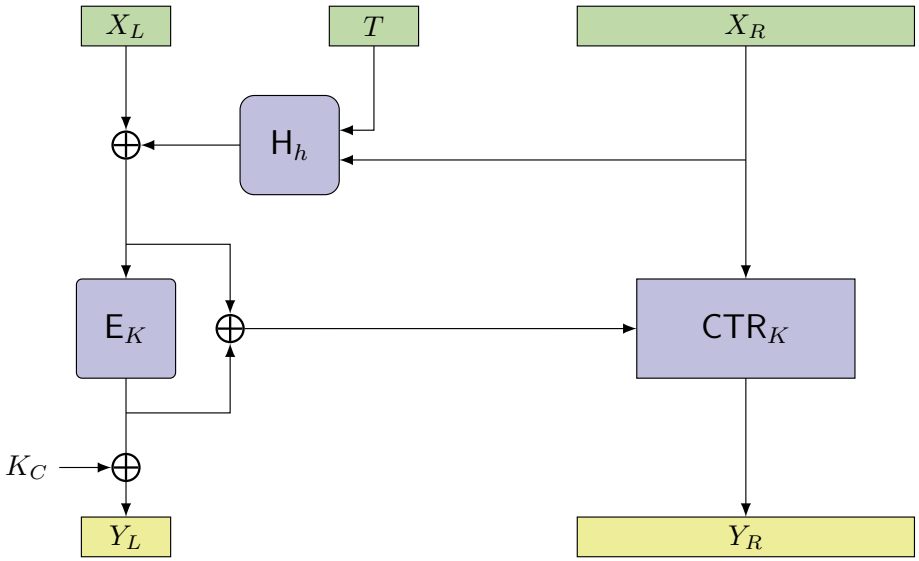


RPRP secure

- HCTR construction [WFW05] as a base.
- HEC:
  1. remove lower layer
  2. add a masking key K_C

HEC Construction

HEC (Hash-Encipher-Counter)

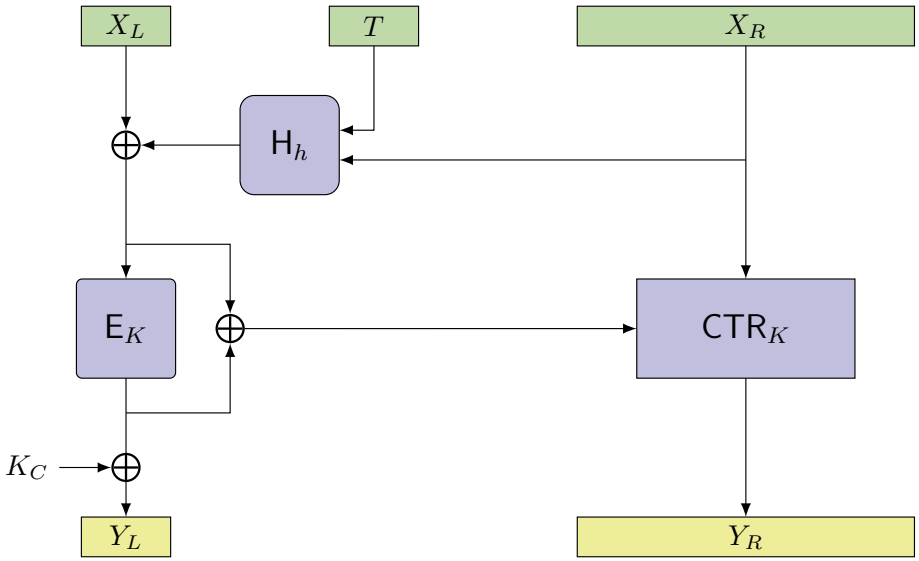


RPRP secure

- HCTR construction [WFW05] as a base.
- HEC:
  1. remove lower layer
  2. add a masking key K_C
- Difference to HCTR:
  1. One less pass over X_R.

HEC Construction

HEC (Hash-Encipher-Counter)

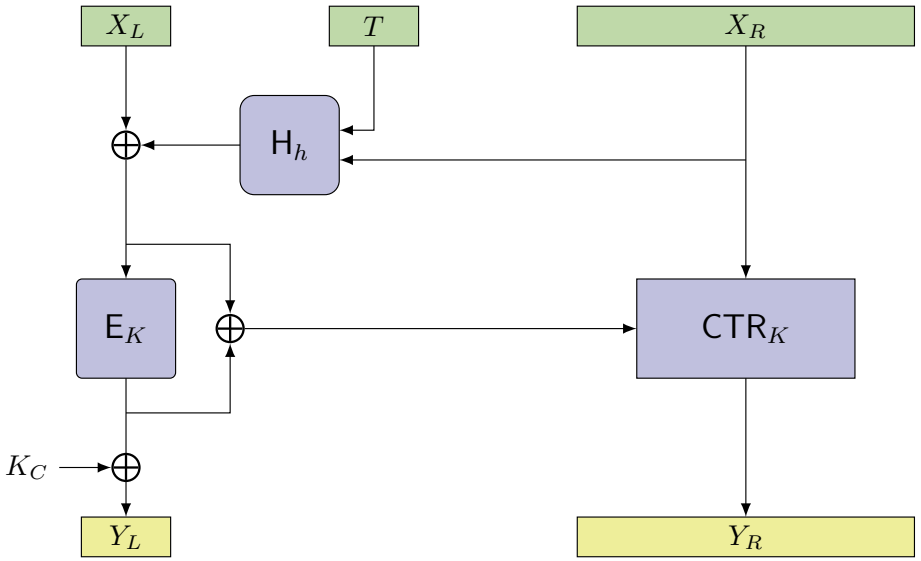


RPRP secure

- HCTR construction [WFW05] as a base.
- HEC:
  1. remove lower layer
  2. add a masking key K_C
- Difference to HCTR:
  1. One less pass over X_R.
  2. One additional key (but no key-scheduling).

HEC Construction

HEC (Hash-Encipher-Counter)



RPRP secure

- HCTR construction [WFW05] as a base.
- HEC:
  1. remove lower layer
  2. add a masking key K_C
- Difference to HCTR:
  1. One less pass over X_R.
  2. One additional key (but no key-scheduling).
- H-Coefficient technique proof with birthday bound security.

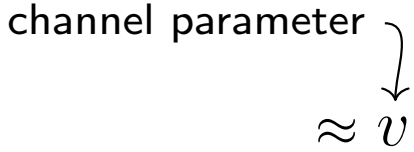
Extending the Left Domain

Motivation for Domain Extension

- Integrity term in the security bound when building order-resilient channels from an RPRP:

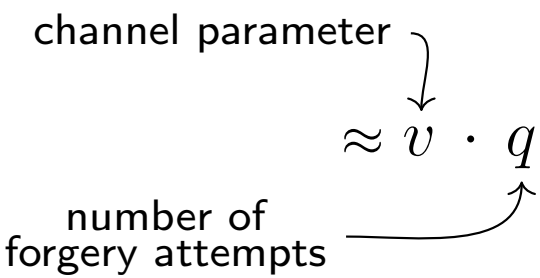
Motivation for Domain Extension

- Integrity term in the security bound when building order-resilient channels from an RPRP:



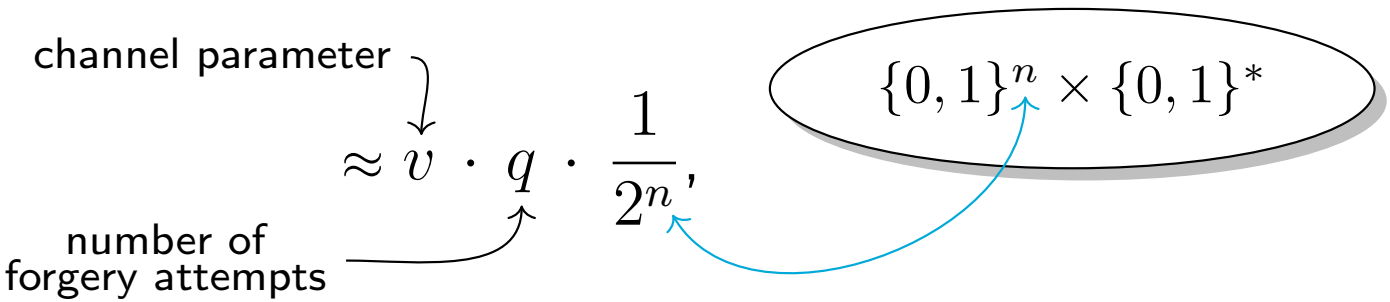
Motivation for Domain Extension

- Integrity term in the security bound when building order-resilient channels from an RPRP:



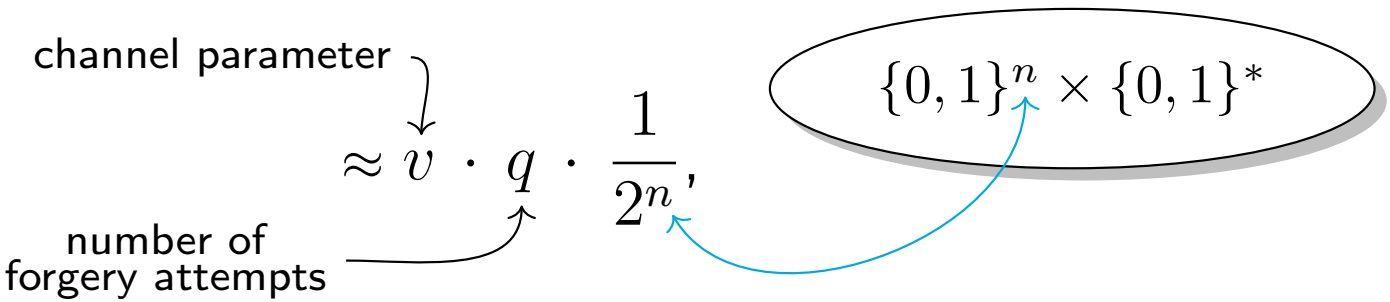
Motivation for Domain Extension

- Integrity term in the security bound when building order-resilient channels from an RPRP:



Motivation for Domain Extension

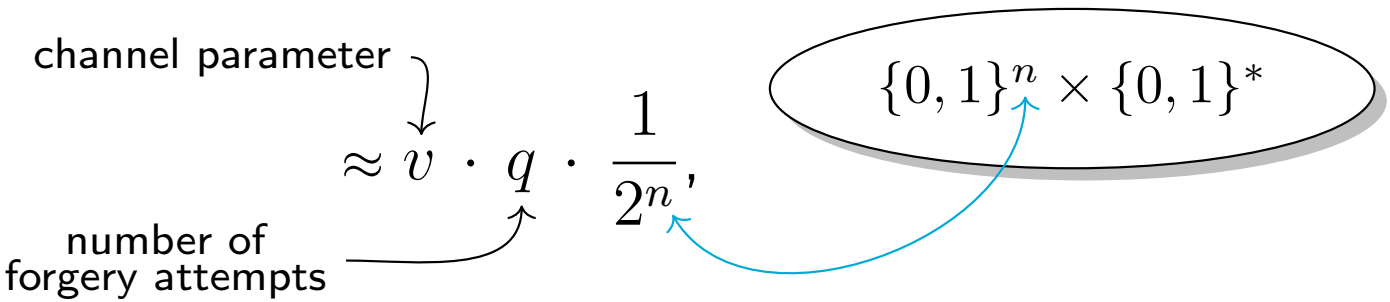
- Integrity term in the security bound when building order-resilient channels from an RPRP:



1. n usually corresponds to blocksize b of some (tweakable) blockcipher $\rightarrow n = 128$.

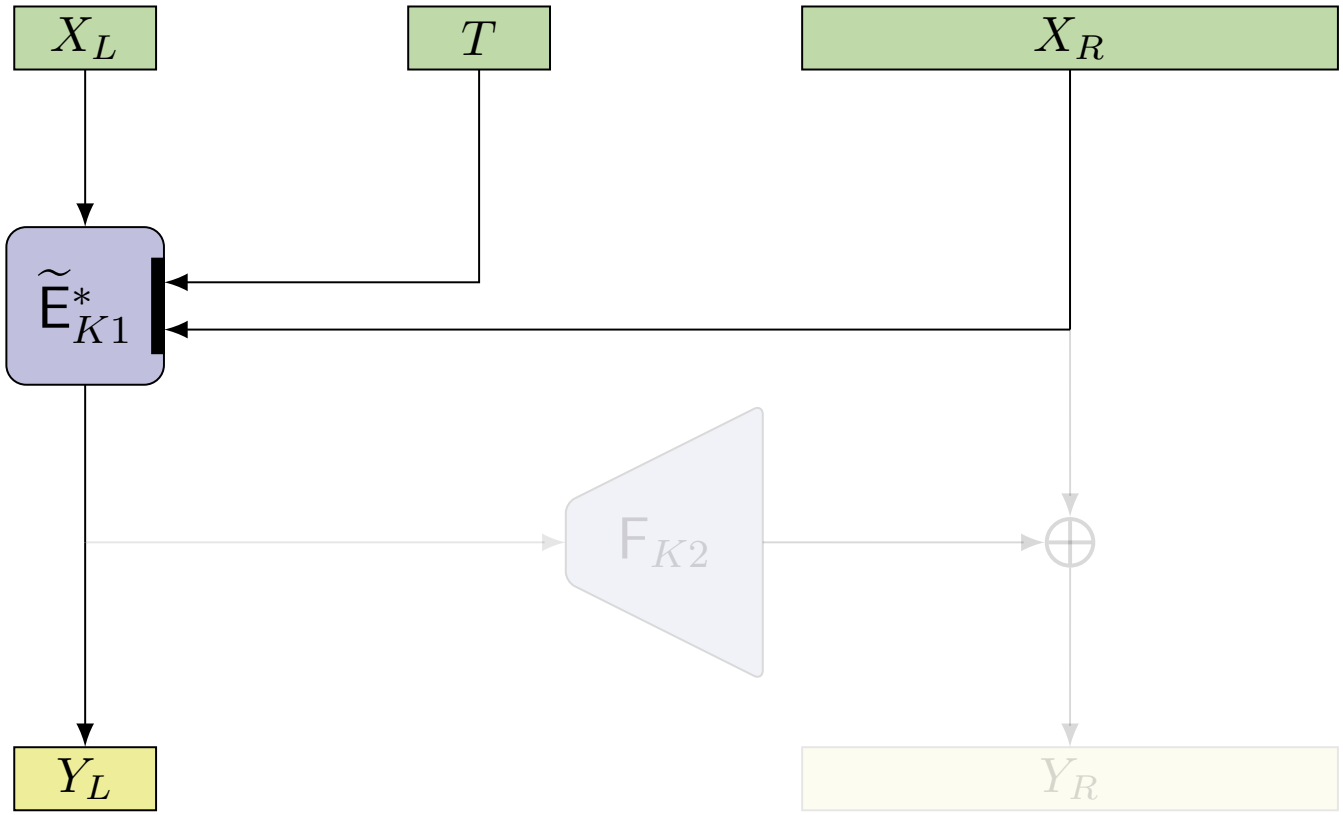
Motivation for Domain Extension

- Integrity term in the security bound when building order-resilient channels from an RPRP:

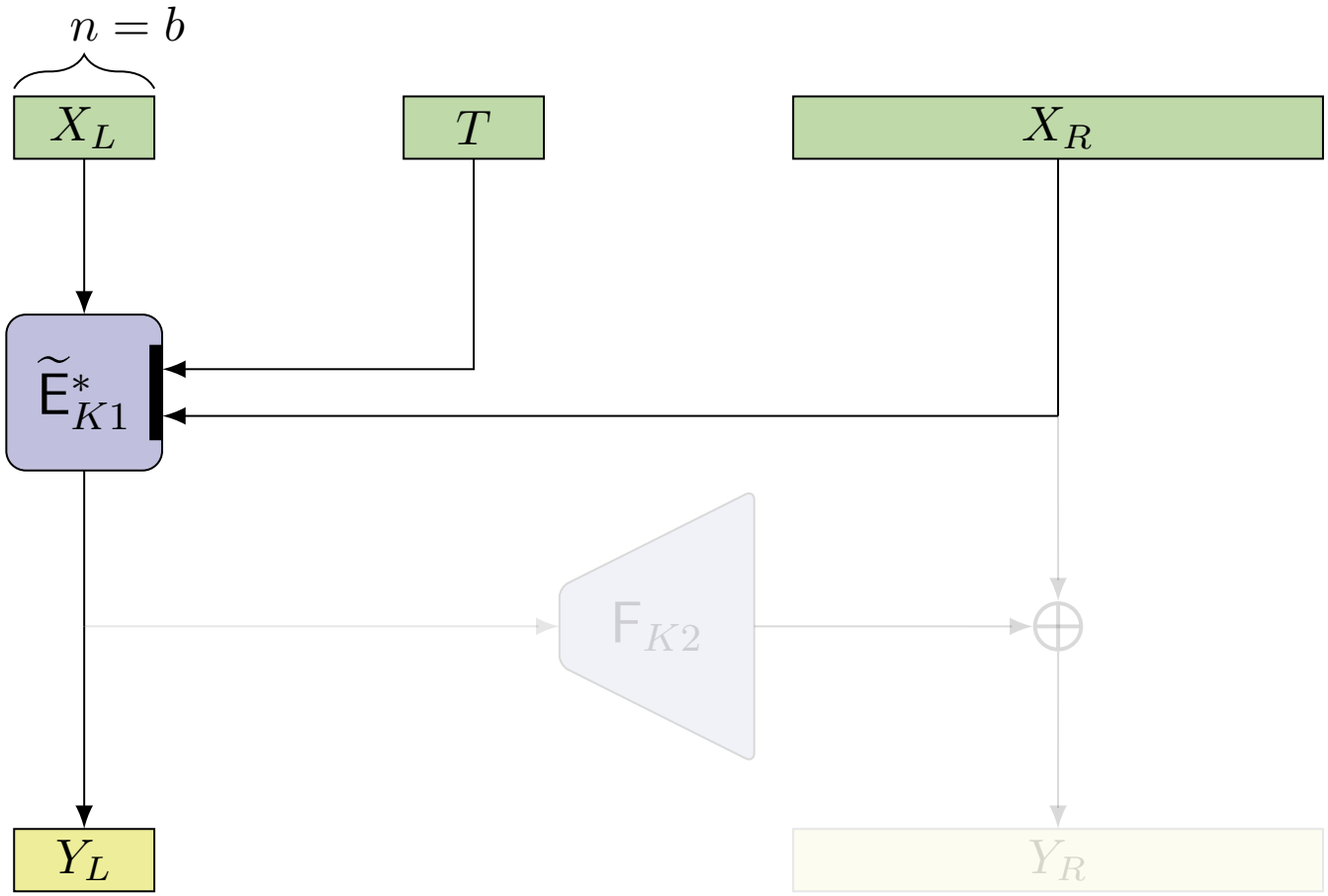


1. n usually corresponds to blocksize b of some (tweakable) blockcipher $\rightarrow n = 128$.
2. In certain cases both v and q can be large (e.g., up to 2^{64}), so $\frac{vq}{2^{128}}$ may not be enough.

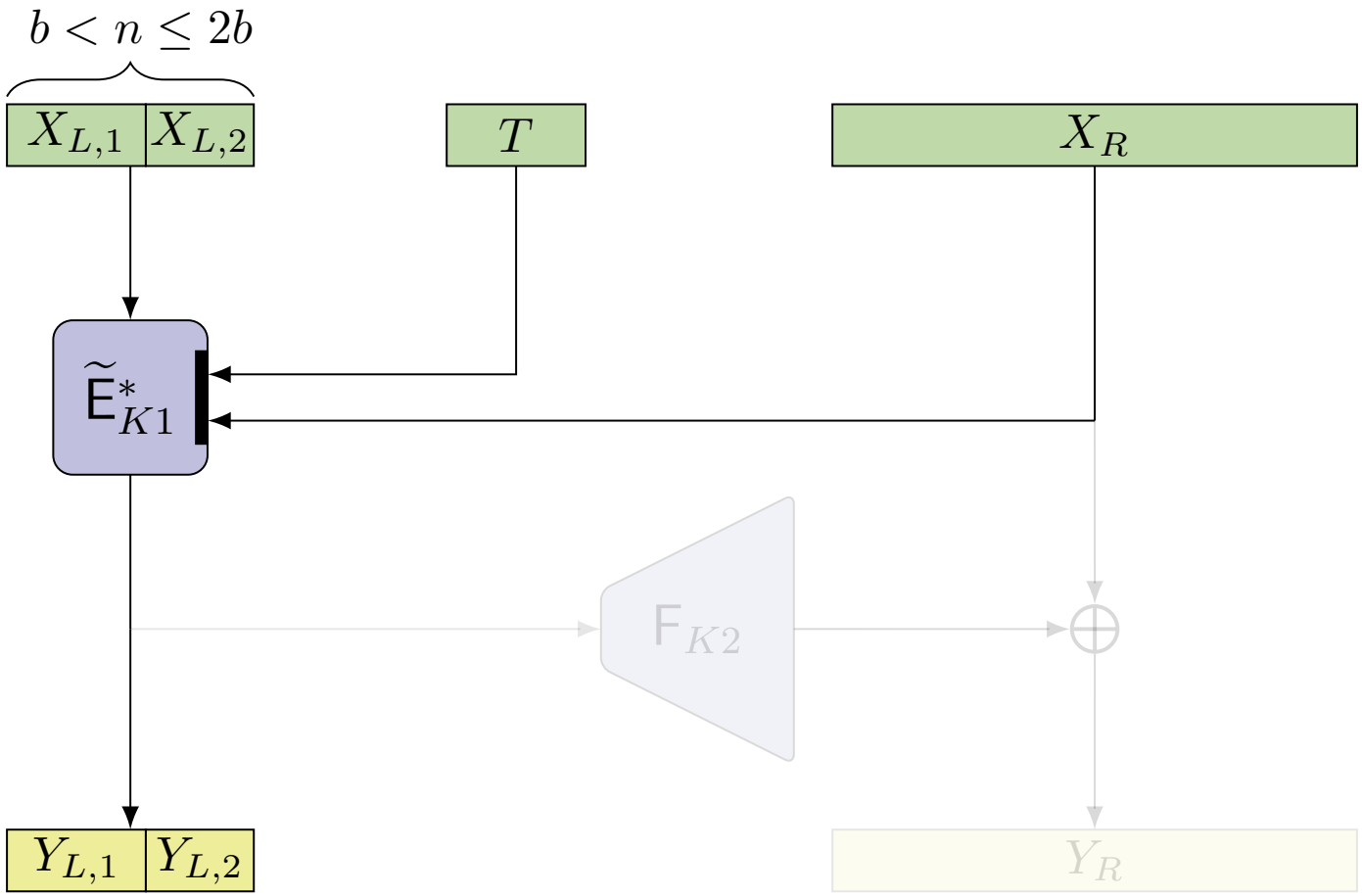
Extending the left domain of UIV [DK22]



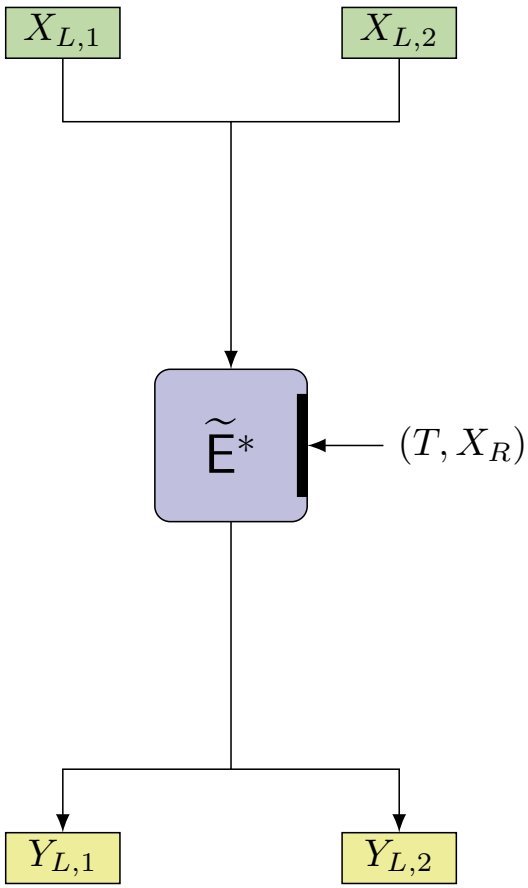
Extending the left domain of UIV [DK22]



Extending the left domain of UIV [DK22]

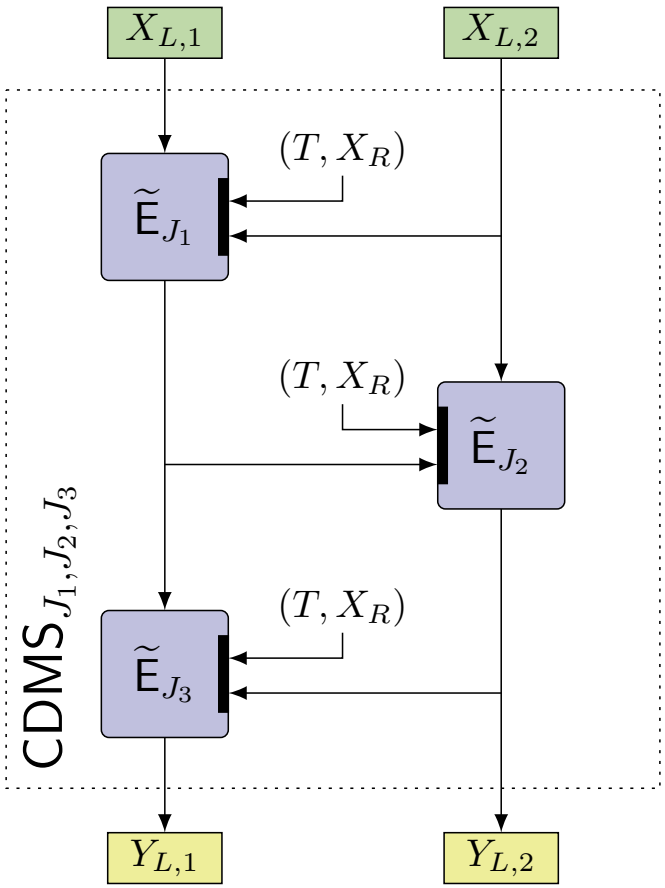


Extending the left domain of UIV: $n = 2b$



Extend left domain $b \to 2b$.

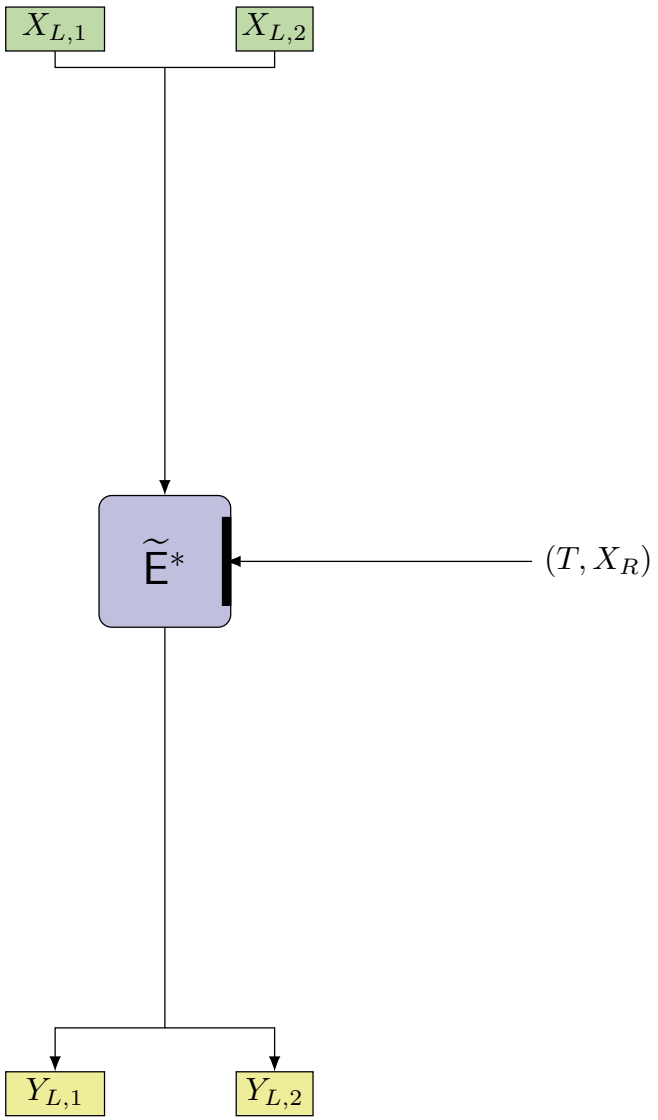
Extending the left domain of UIV: $n = 2b$



Extend left domain $b \to 2b$.

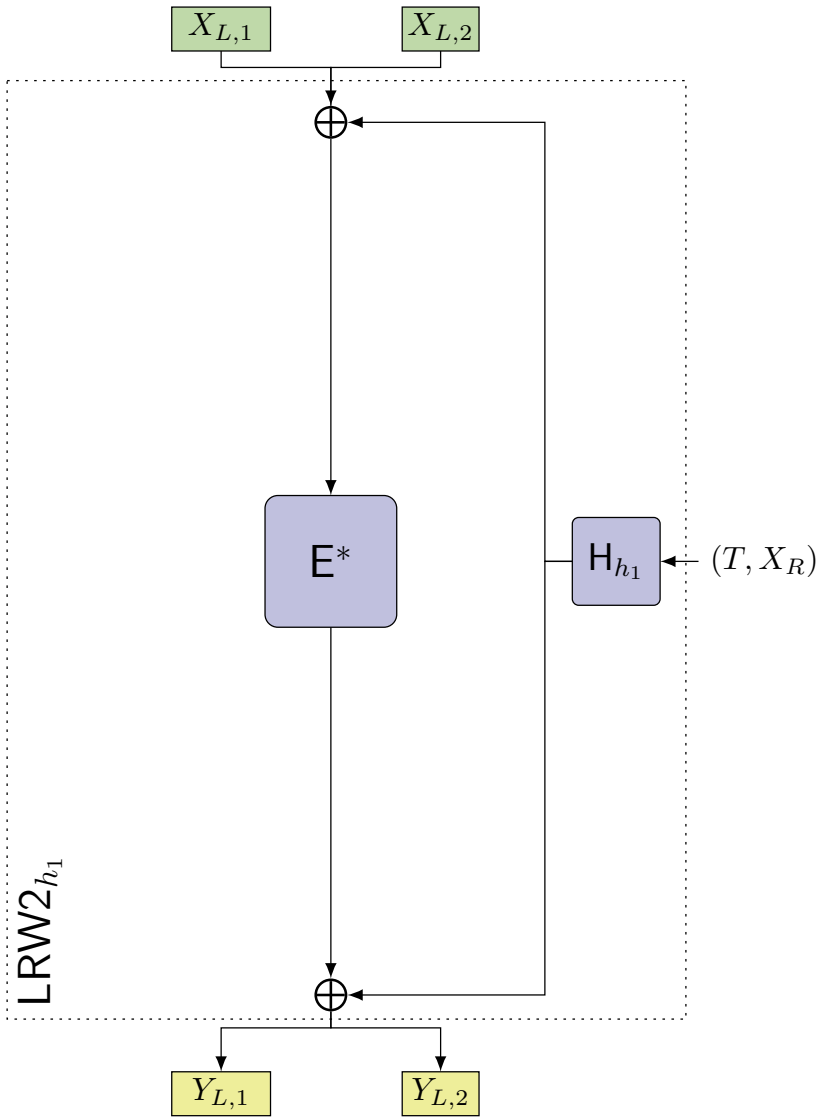
- CDMS by Coron et al. [CDMS10].
- $2b$-bit TBC $\widetilde{\mathsf{E}}^*$ from b-bit TBC $\widetilde{\mathsf{E}}$

Extending the left domain of UIV: $b < n < 2b$



Extend left domain $b \to b + s$,
where $1 \leq s < b$.

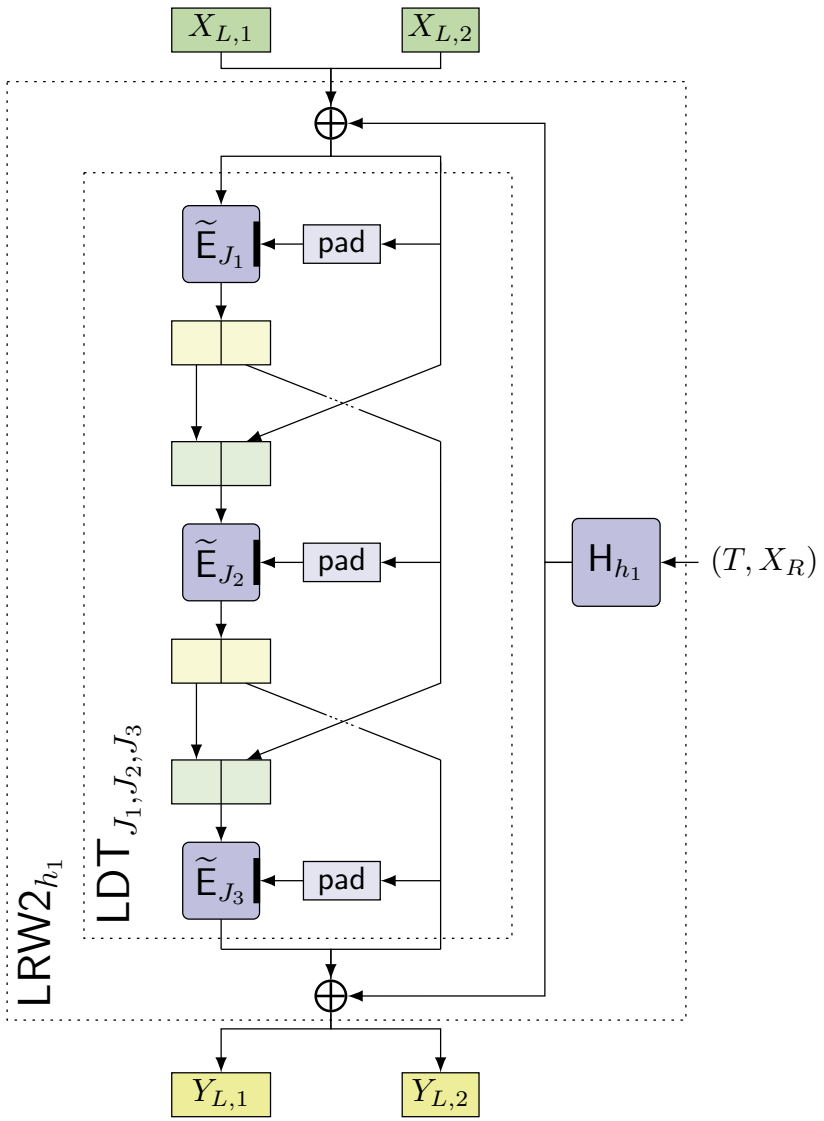
Extending the left domain of UIV: $b < n < 2b$



Extend left domain $b \to b + s$, where $1 \leq s < b$.

1. LRW2 instantiation [LRW11] of $\widetilde{\mathsf{E}}^*$ using:
   - AXU hash function H
   - blockcipher E^* of block size $b + s$

Extending the left domain of UIV: $b < n < 2b$



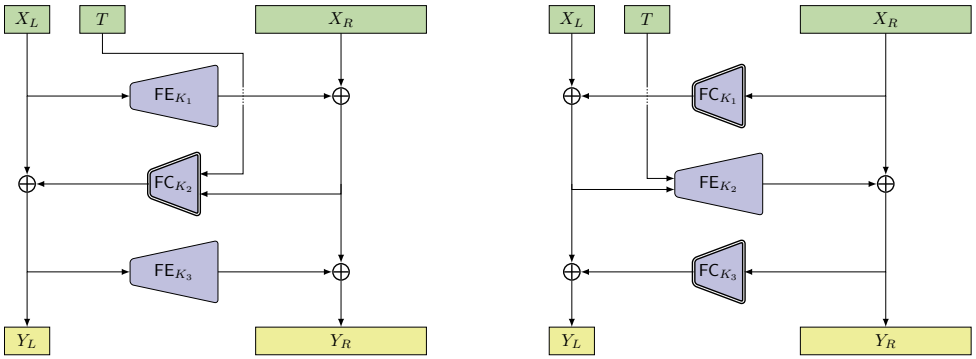
Extend left domain $b \to b + s$, where $1 \leq s < b$.

1. LRW2 instantiation [LRW11] of $\widetilde{\mathsf{E}}^*$ using:
   - AXU hash function H
   - blockcipher E^* of block size $b + s$
2. LDT instantiation [CMN18] of E^* using:
   - tweakable blockcipher of block size b

Extending the left domain of ECE / CEC and HEC?

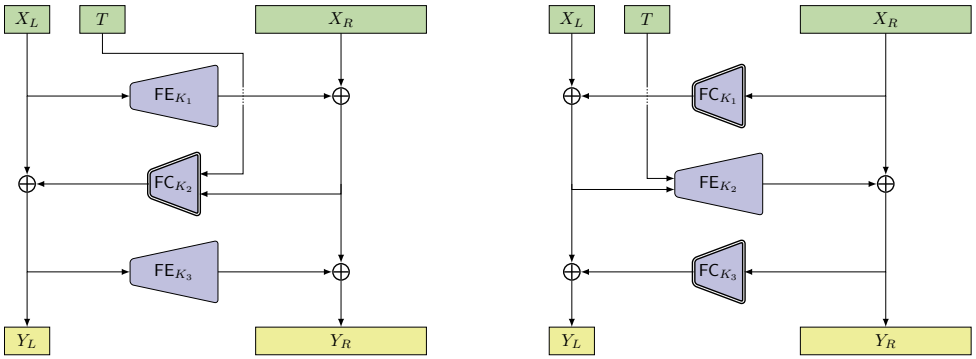
Extending the left domain of ECE / CEC and HEC?

ECE / CEC:



Extending the left domain of ECE / CEC and HEC?

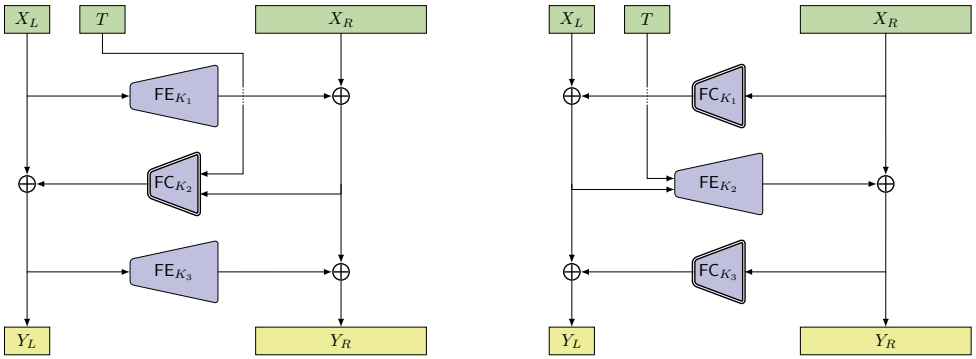
ECE / CEC:



- Extending the left domain of an unbalanced Feistel scheme is usually easy.

Extending the left domain of ECE / CEC and HEC?

ECE / CEC:

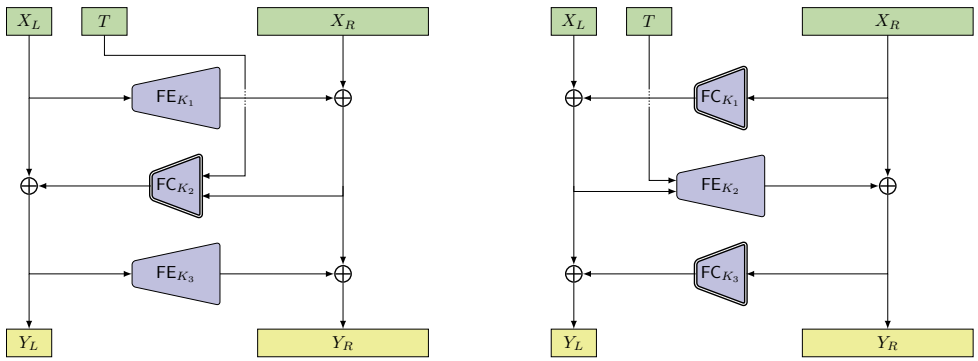


- Extending the left domain of an unbalanced Feistel scheme is usually easy.

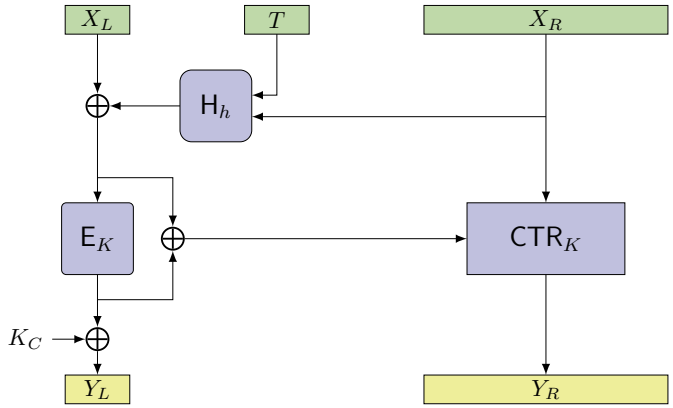
$\rightarrow$ Adjust the input/output size of the expanding and compressing PRF.

Extending the left domain of ECE / CEC and HEC?

ECE / CEC:



HEC:

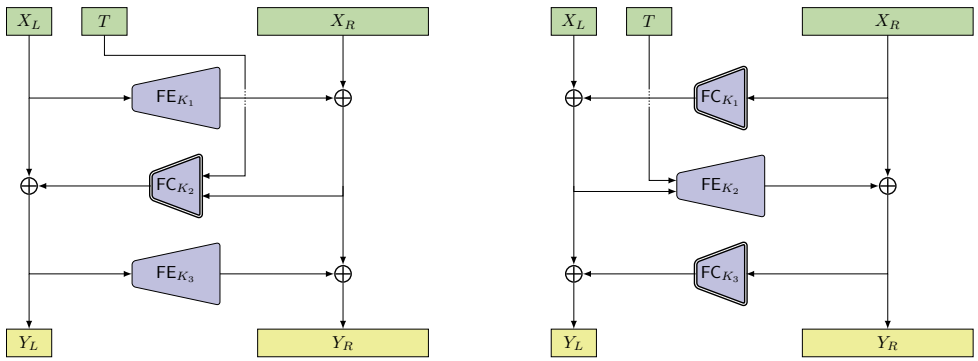


- Extending the left domain of an unbalanced Feistel scheme is usually easy.

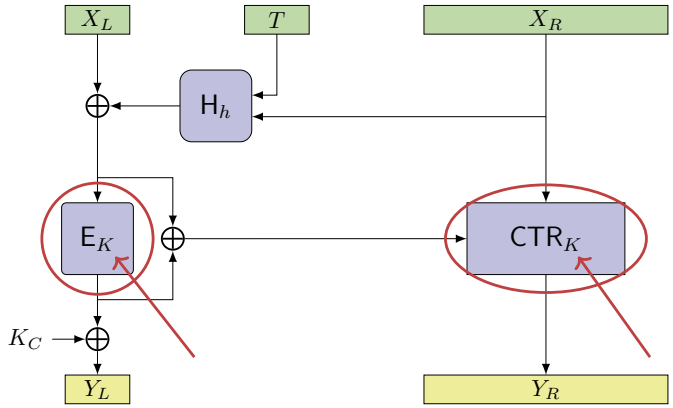
$\rightarrow$ Adjust the input/output size of the expanding and compressing PRF.

Extending the left domain of ECE / CEC and HEC?

ECE / CEC:



HEC:



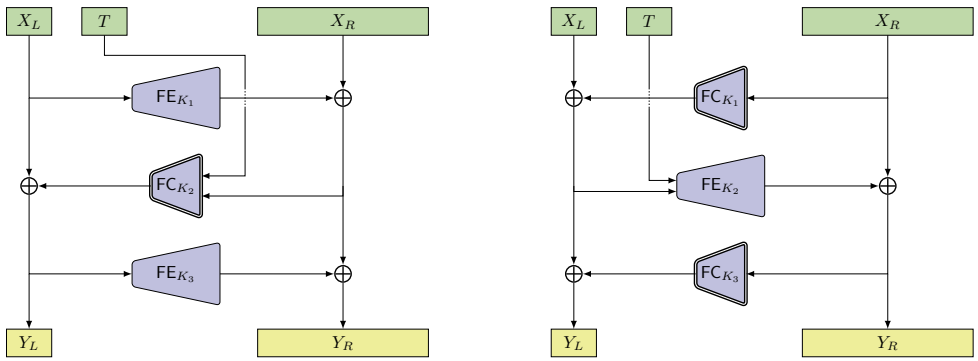
- Extending the left domain of an unbalanced Feistel scheme is usually easy.

→ Adjust the input/output size of the expanding and compressing PRF.

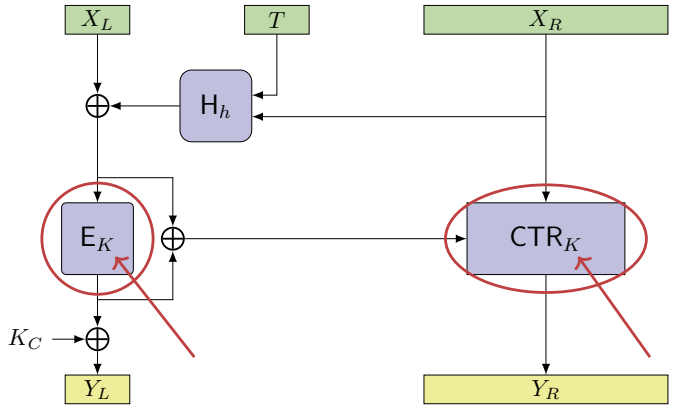
- "Left" and "right" component not independent → black-box solution not possible.

Extending the left domain of ECE / CEC and HEC?

ECE / CEC:



HEC:



- Extending the left domain of an unbalanced Feistel scheme is usually easy.

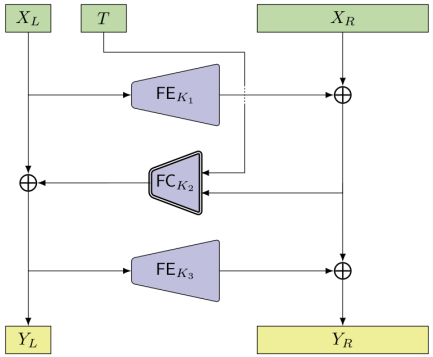
→ Adjust the input/output size of the expanding and compressing PRF.

- "Left" and "right" component not independent → black-box solution not possible.
- Future work: dedicated domain extender for HEC.

Summary

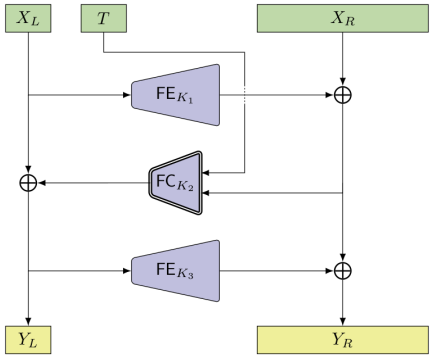
Summary

ECE (RPRPd-secure)

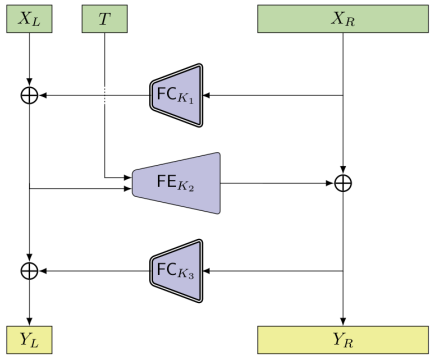


Summary

ECE (RPRPd-secure)

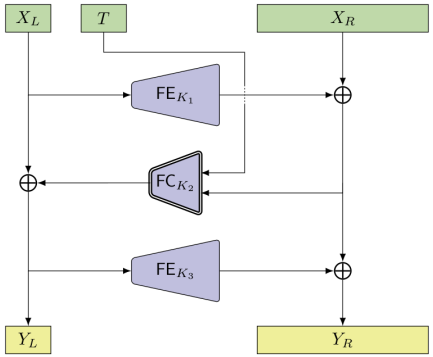


CEC (RPRPg-secure)

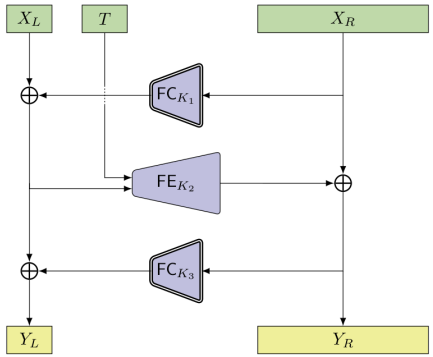


Summary

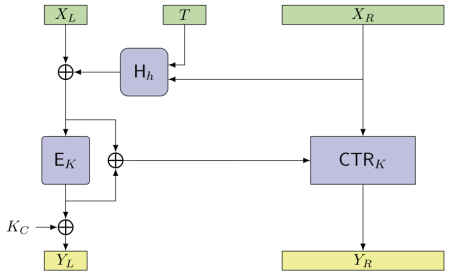
ECE (RPRPd-secure)



CEC (RPRPg-secure)

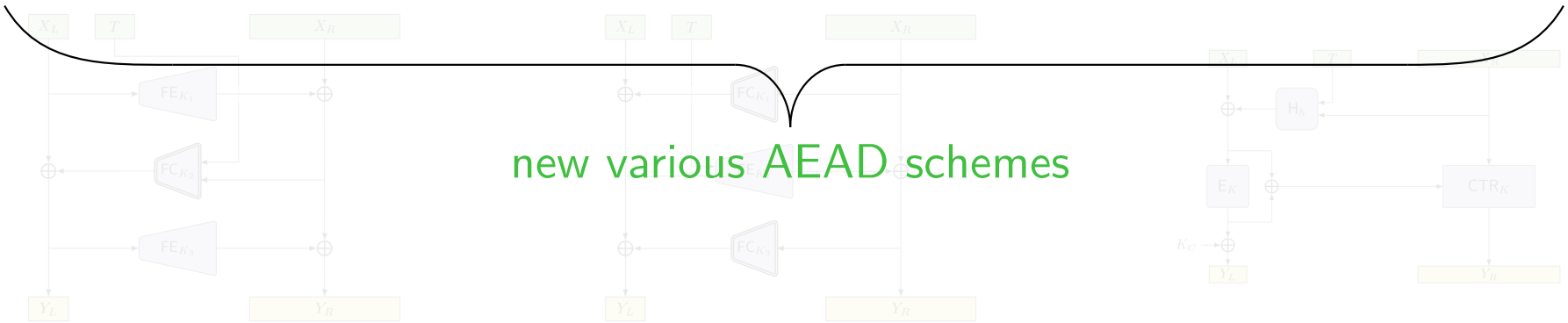


HEC (RPRP-secure)



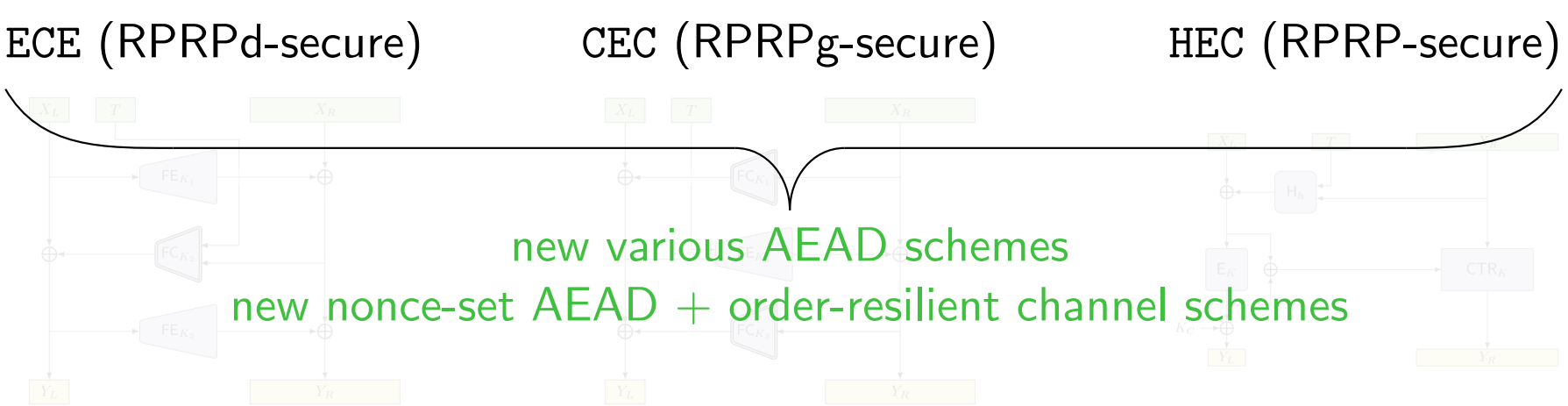
Summary

ECE (RPRPd-secure) CEC (RPRPg-secure) HEC (RPRP-secure)

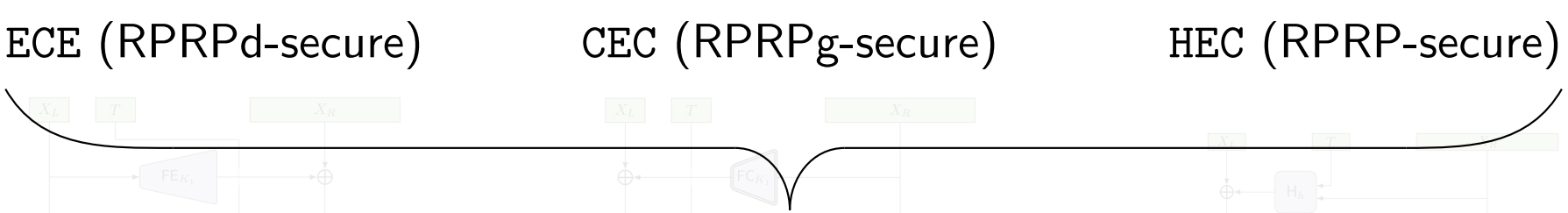


new various AEAD schemes

Summary


ECE (RPRPd-secure)
CEC (RPRPg-secure)
HEC (RPRP-secure)
new various AEAD schemes
new nonce-set AEAD + order-resilient channel schemes

Summary

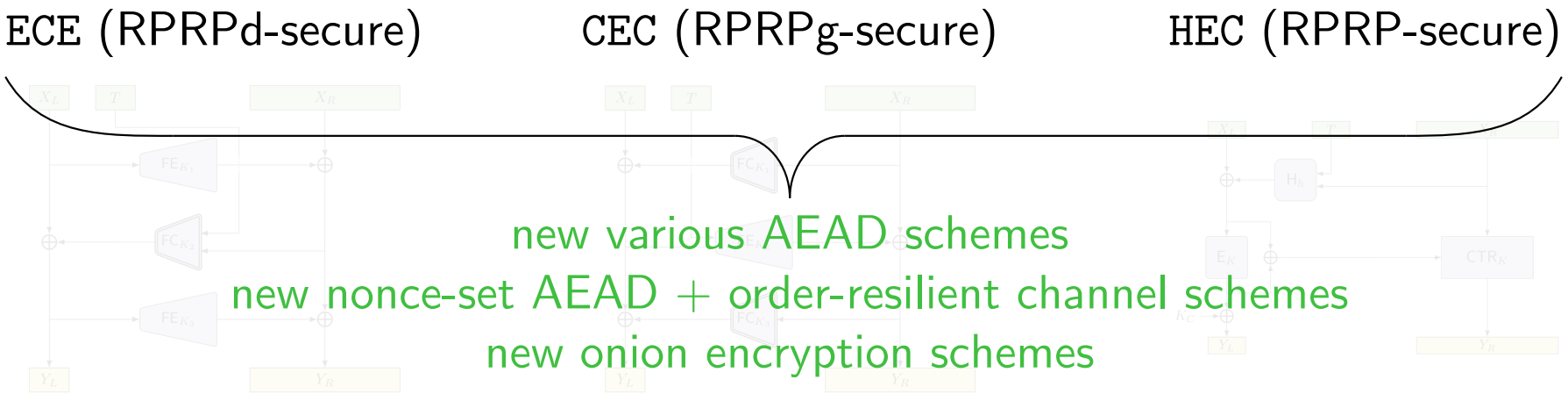


new various AEAD schemes

new nonce-set AEAD + order-resilient channel schemes

new onion encryption schemes

Summary



RPRPs are indeed natural!

Summary

ECE (RPRPd-secure) CEC (RPRPg-secure) HEC (RPRP-secure)

new various AEAD schemes
new nonce-set AEAD + order-resilient channel schemes
new onion encryption schemes

RPRPs are indeed natural!

ia.cr/2023/1432

Thank you for listening.

Questions?

References I

[BDHAK22] Norica Bacuieti, Joan Daemen, Seth Hoffert, Gilles Van Assche, and Ronny Van Keer. "Jammin' on the Deck". In: *ASIACRYPT 2022, Part II*. Ed. by Shweta Agrawal and Dongdai Lin. Vol. 13792. LNCS. Springer, Heidelberg, Dec. 2022, pp. 555–584. DOI: 10.1007/978-3-031-22966-4_19.

[BNT19] Mihir Bellare, Ruth Ng, and Björn Tackmann. "Nonces Are Noticed: AEAD Revisited". In: *CRYPTO 2019, Part I*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11692. LCNS. Springer, Heidelberg, Aug. 2019, pp. 235–265. DOI: 10.1007/978-3-030-26948-7_9.

[BR00] Mihir Bellare and Phillip Rogaway. "Encode-Then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography". In: *ASIACRYPT 2000*. Ed. by Tatsuaki Okamoto. Vol. 1976. LCNS. Kyoto, Japan: Springer, Heidelberg, Dec. 2000, pp. 317–330. DOI: 10.1007/3-540-44448-3_24.

[CDMS10] Jean-Sébastien Coron, Yevgeniy Dodis, Avradip Mandal, and Yannick Seurin. "A Domain Extender for the Ideal Cipher". In: *TCC 2010*. Ed. by Daniele Micciancio. Vol. 5978. LCNS. Springer, Heidelberg, Feb. 2010, pp. 273–289. DOI: 10.1007/978-3-642-11799-2_17.

References II

[CMN18] Yu Long Chen, Bart Mennink, and Mridul Nandi. "Short Variable Length Domain Extenders with Beyond Birthday Bound Security". In: *ASIACRYPT 2018, Part I*. Ed. by Thomas Peyrin and Steven Galbraith. Vol. 11272. LCNS. Springer, Heidelberg, Dec. 2018, pp. 244–274. DOI: 10.1007/978-3-030-03326-2_9.

[DK22] Jean Paul Degabriele and Vukašin Karadžić. "Overloading the Nonce: Rugged PRPs, Nonce-Set AEAD, and Order-Resilient Channels". In: *CRYPTO 2022, Part IV*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13510. LNCS. Springer, Heidelberg, Aug. 2022, pp. 264–295. DOI: 10.1007/978-3-031-15985-5_10.

[DKMMS22] Jean Paul Degabriele, Vukašin Karadžić, Alessandro Melloni, Jean-Pierre Münch, and Martijn Stam. *Rugged Pseudorandom Permutations and Their Applications*. Presented at the IACR Real World Crypto Symposium 2022. 2022.

[FL09] Xinwen Fu and Zhen Ling. "One cell is enough to break Tor's anonymity". In: *Proceedings of Black Hat DC 2009* (2009).

[LR88] Michael Luby and Charles Rackoff. "How to construct pseudorandom permutations from pseudorandom functions". In: *SIAM Journal on Computing* 17.2 (1988).

References III

[LRW11] Moses Liskov, Ronald L. Rivest, and David Wagner. "Tweakable Block Ciphers". In: *Journal of Cryptology* 24.3 (July 2011), pp. 588–613. DOI: 10.1007/s00145-010-9073-y.

[WFW05] Peng Wang, Dengguo Feng, and Wenling Wu. "HCTR: A Variable-Input-Length Enciphering Mode". In: *Information Security and Cryptology*. Ed. by Dengguo Feng, Dongdai Lin, and Moti Yung. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 175–188.