Rugged Pseudorandom Permutations and Their Applications

Jean Paul Degabriele and Vukašin Karadžić

Technology Innovation Institute
TECHNISCHE UNIVERSITÄT DARMSTADT

Outline

Cryptography Research Centre

- The Security Definition
- Transforming Rugged PRPs into AEAD
- Nonce-Set AEAD and Order-Resilient Channels
- Application to Onion Encryption in Tor
- RPRP Constructions

The Security Definition

The Security of Variable-Length Tweakable Ciphers

PRP | RPRP | SPRP

Is there something useful in between?

E vs Π

$E, f(D)$ vs $\Pi, f(\Pi^{-1})$

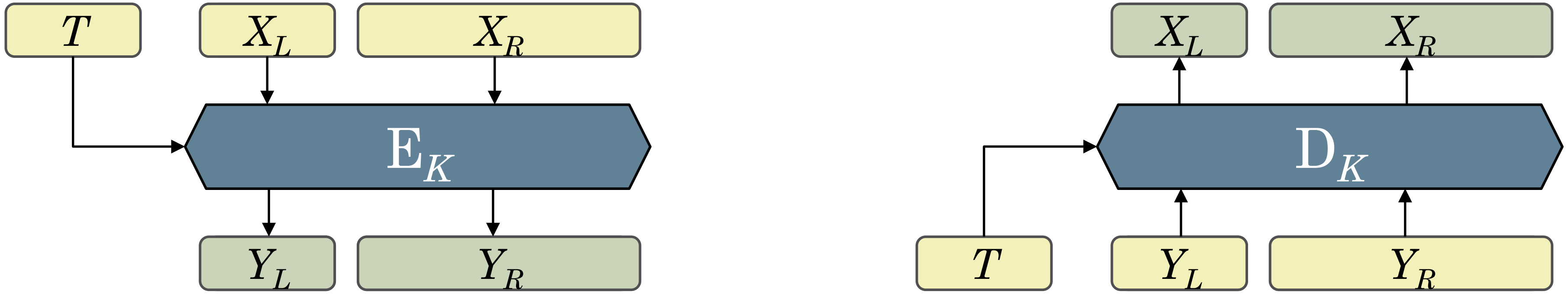
E, D vs Π, Π^{-1}

- Not very useful as most applications require deciphering

- Intuitively f( ) limits access to D and Π^{-1}
- Goal: Notion permitting **more efficient** constructions that have **practical** applications.

- Strong security, but
- Heavy Constructions
- Or require stronger assumptions (AEZ)
- Sometimes Overkill

Rugged Pseudorandom Permutations

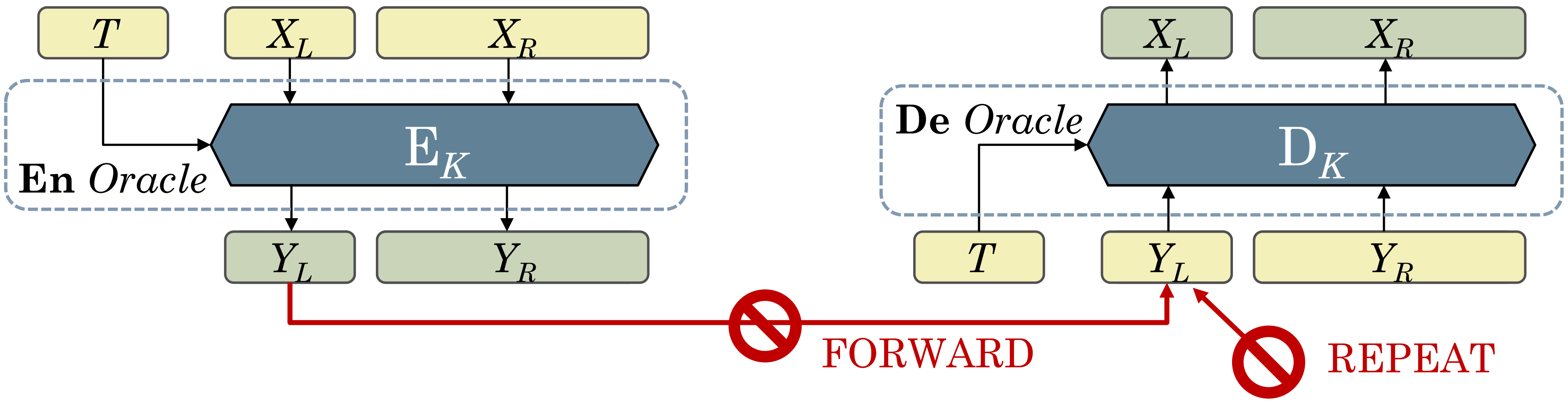


- The notion of a Rugged PRP requires a slightly more stringent syntax.

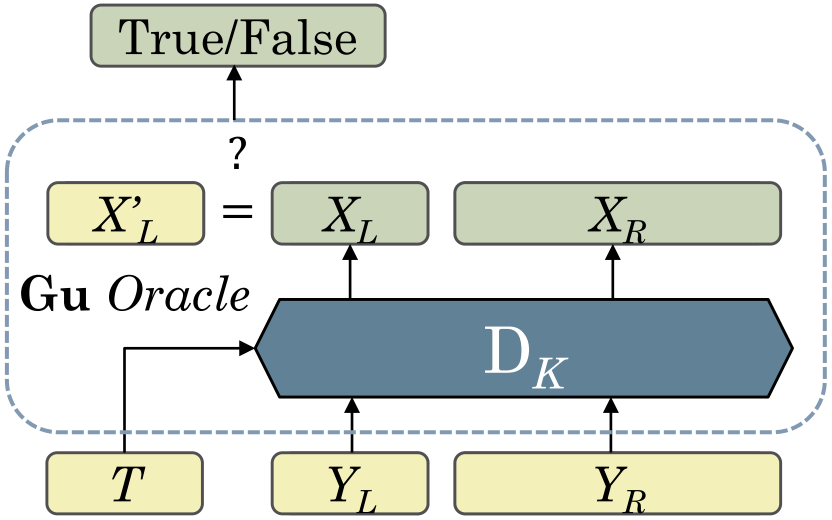
- Namely the **(VIL) tweakable cipher** must operate over a **split domain,** such as $\{0,1\}^n \times \{0,1\}^*$, where n is in the range128-256 bits.

Rugged Pseudorandom Permutations

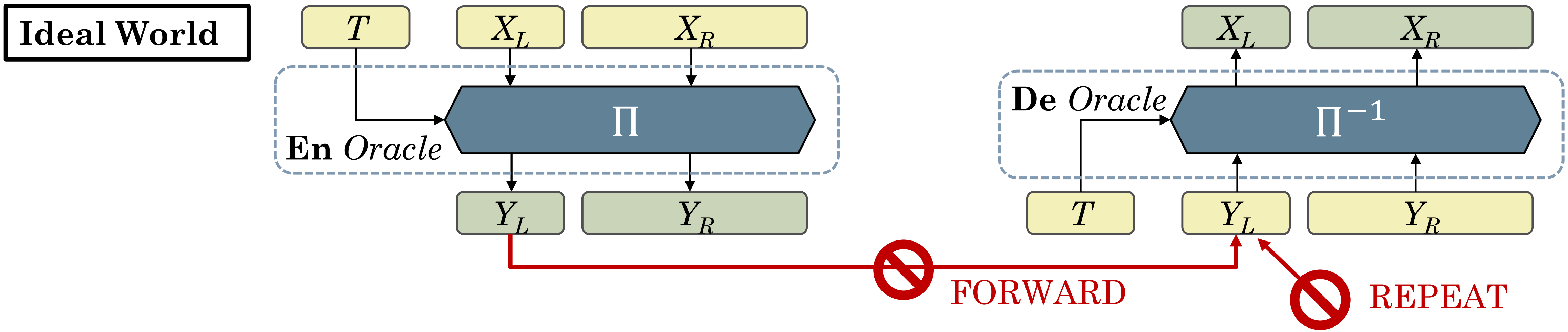
Real World



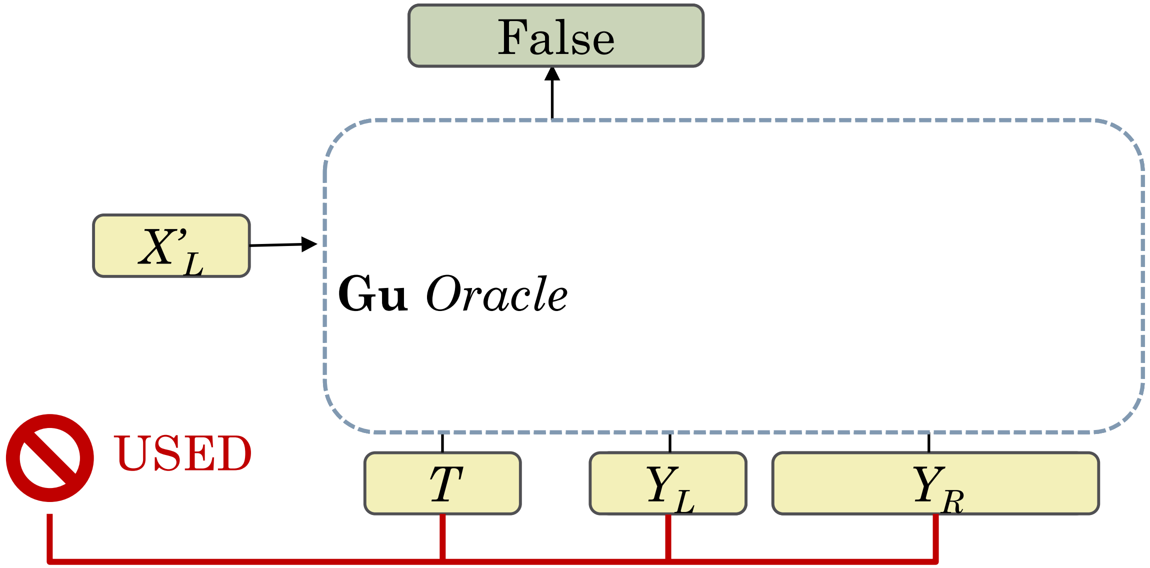
- Deciphering can be accessed via two separate oracles:
- **De** - restricted queries, full output.
- **Gu** – unrestricted queries, 1-bit output.



Rugged Pseudorandom Permutations

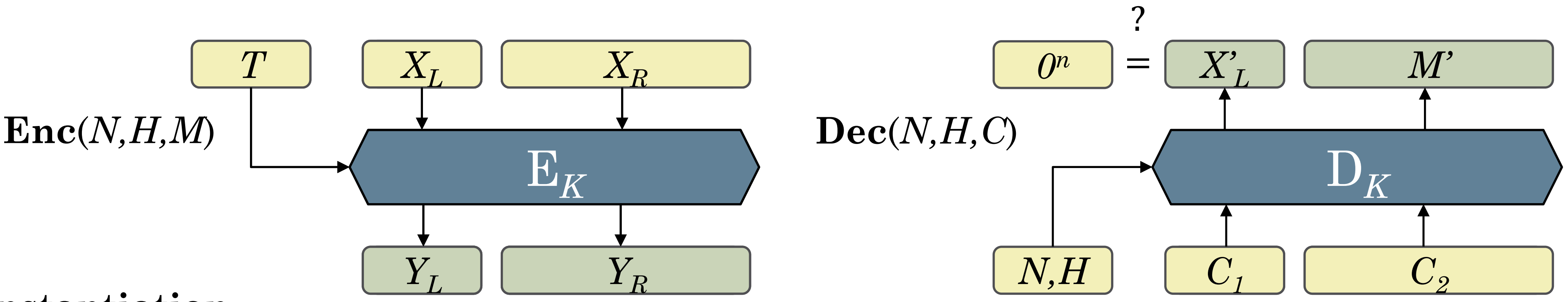


- Replace E_K with an ideal cipher Π.
- $\mathbf{Gu}$ always returns false.
- For satisfiability $\mathbf{Gu}$ queries must **not** be **trivial to guess**.



Transforming RPRPs into AEAD

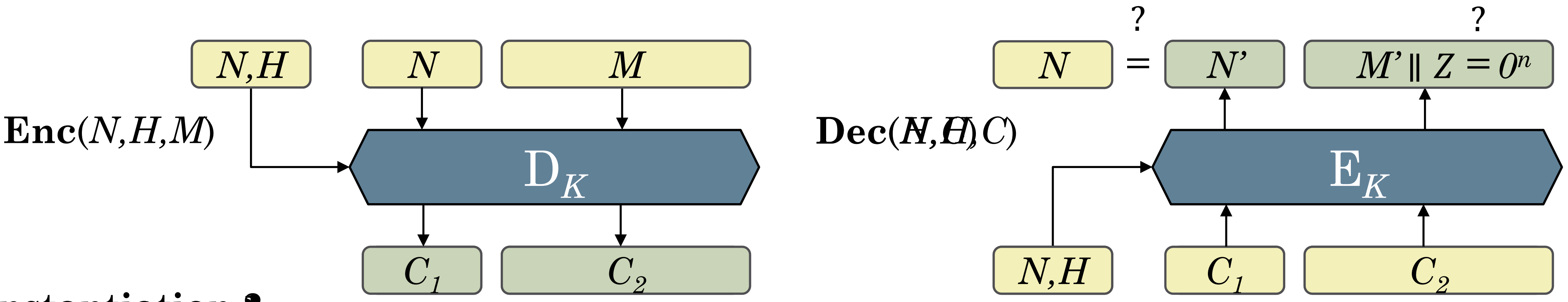
The EtE Transform



EtE instantiation

- We revisit and adapt the **Encode-then-Encipher paradigm** [BelRog00, ShrTer13] in the context of RPRPs.
- EtE is slightly more general, the above is a specific instantiation of it.
- $(\mathbf{E}_K, \mathbf{D}_K)$ is RPRP secure $\Longrightarrow$ EtE is **Misuse-Resistant AEAD**.

The EtD Transform

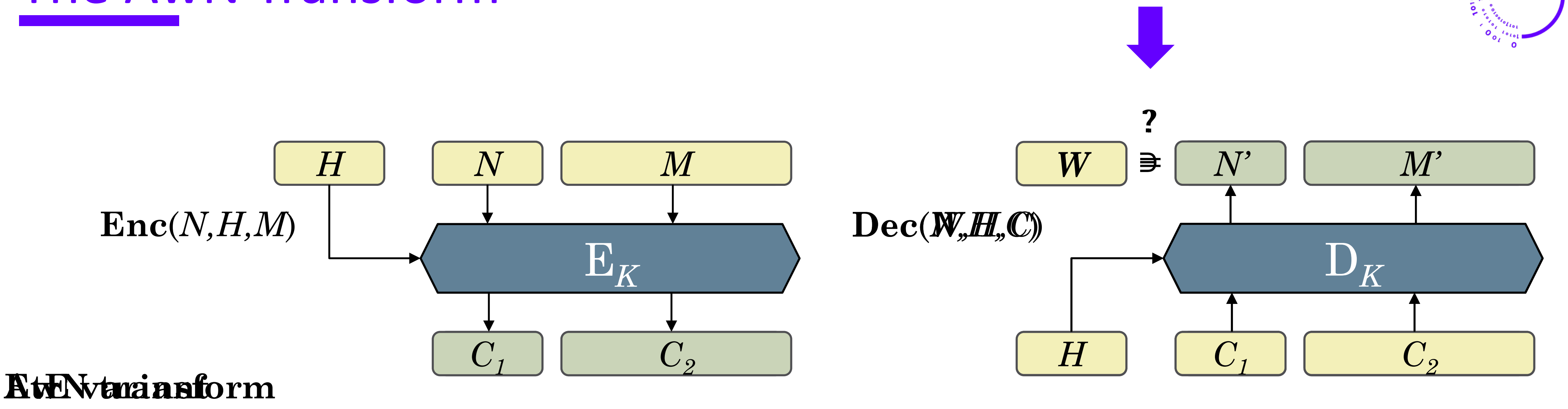


EtD instantiation 2

- $(\mathbf{E}_K, \mathbf{D}_K)$ is RPRP secure $\Longrightarrow$ EtD yields a **RUPAE nonce-hiding AEAD**.
- However we can instantiate it differently to reduce the ciphertext expansion by using the **nonce to authenticate** the ciphertext.
- $(\mathbf{E}_K, \mathbf{D}_K)$ is RPRP secure $\Longrightarrow$ EtD is a (standard) **AEAD** that is **RUPAE** secure**.**

Nonce-Set AEAD and Order-Resilient Channels

The AwN Transform



- We can also use the **nonce to authenticate** in the **EtE** transform and obtain a nonce-hiding AEAD.
- We can generalize this further by **testing the nonce for set membership** instead of equality, yielding the **AwN** transform.
- **AwN** transforms an RPRP into a **Nonce-Set AEAD** that is **Misuse-Resistant**.

Why Nonce-Set AEAD?

Cryptography Research Centre

- Nonce-Set AEAD serves as a **stepping stone** for realizing **order-resilient channels** such as **QUIC** and **DTLS**.

- Several possibilities arise for handling **reorderings**, **replays**, **modifications**, and **deletions**, and how much of each to tolerate.

- Typical constructions employ one or more **window mechanisms**, which add complexity—making them **hard to understand and analyze**.

- In general, it is unclear how these **additional mechanisms** interact with AEAD and what the **overall security** of the channel is.

The Support Predicate

- The various functionalities of such channels can be formally characterized by a **support predicate**:

$$\boldsymbol{accept/reject \leftarrow supp(C, C_S, DC_R)}$$

- It was developed in [Bac19, FGJ20] as a **generalization** of the **silencing approach** by [RogZha18].

- The support predicate permeates into all aspects of the secure channel **correctness**, **security**, and **robustness** [FGJ20].

Order-Resilient Channels from NS-AEAD

Init()

$(\mathrm{st}_s, \mathrm{st}_r) \xleftarrow{\$} \mathsf{StInit}()$

$K \xleftarrow{\$} \{0,1\}^k$

$\mathrm{stk}_s \leftarrow (\mathrm{st}_s, K)$

$\mathrm{stk}_r \leftarrow (\mathrm{st}_r, K)$

$\mathbf{return}\ (\mathrm{stk}_s, \mathrm{stk}_r)$

Send(stk_s, A, M)

$(\mathrm{st}_s, K) \leftarrow \mathrm{stk}_s$

$(\mathrm{st}'_s, N) \leftarrow \mathsf{NonceExtract}(\mathrm{st}_s)$

$\mathbf{if}\ N = \bot\ \mathbf{then}$

$\quad \mathbf{return}\ (\mathrm{st}'_s, \bot)$

$C \leftarrow \mathsf{Enc}(K, N, A, M)$

$\mathrm{stk}'_s \leftarrow (\mathrm{st}'_s, K)$

$\mathbf{return}\ (\mathrm{stk}'_s, C)$

Recv(stk_r, A, C)

$(\mathrm{st}_r, K) \leftarrow \mathrm{stk}_r$

$\boldsymbol{W} \leftarrow \mathsf{NonceSetPolicy}(\mathrm{st}_r)$

$(N, M) \leftarrow \mathsf{Dec}(K, \boldsymbol{W}, A, C)$

$\mathbf{if}\ (N, M) = (\bot, \bot)\ \mathbf{then}$

$\quad mn \leftarrow \bot$

$\mathbf{else}$

$\quad (\mathrm{st}'_r, mn) \leftarrow \mathsf{StUpdate}(\mathrm{st}_r, N)$

$\mathrm{stk}'_r \leftarrow (\mathrm{st}'_r, K)$

$\mathbf{return}\ (\mathrm{stk}'_r, mn, M)$

- We present a **universal** and **generic** channel construction from Nonce-Set AEAD for **any desired support predicate**!

- The construction consists of a **Nonce-Set AEAD** (blue) scheme and a tuple of **Nonce-Set Processing (NSP)** scheme (red).

Order-Resilient Channels from NS-AEAD

- We prove this channel construction **correct**, **robust**, and **secure.**

- We only require that the **Nonce-Set AEAD** is secure and that the **NSP scheme** satisfy a functionality property called **faithfulness.**

- Informally, faithfulness says that the **NSP scheme** accurately reproduces the **support predicate logic over the nonces**.

- One can simply **tune the NSP** to the **desired functionality** and plug in their favourite **Nonce-Set AEAD** and **security/robustness** will be **automatic**.

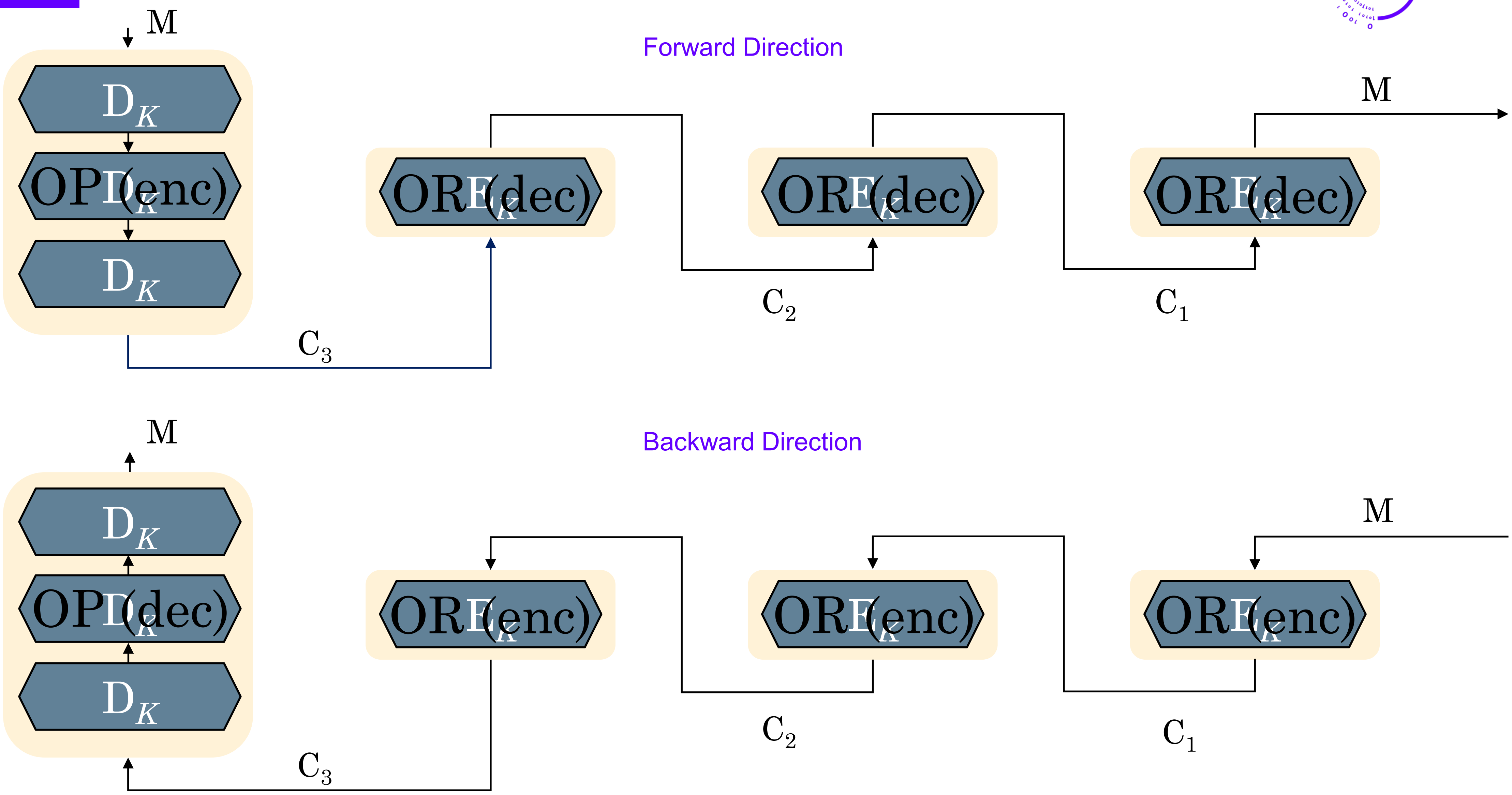
Application to Onion Encryption in Tor

Onion Encryption from RPRPs

Cryptography Research Centre

- Tor is susceptible to **tagging attacks** which undermine privacy by exploiting the **malleability** in its encryption layers.
- To address this, it has been proposed to replace each layer with a **wide-block Strong PRP**, but a **Rugged PRP** turns out to be sufficient.
- With other co-authors we have a proposal for an RPRP-based onion encryption scheme which adds **forward security**, protects against **tagging attacks**, and provides **competitive performance**.

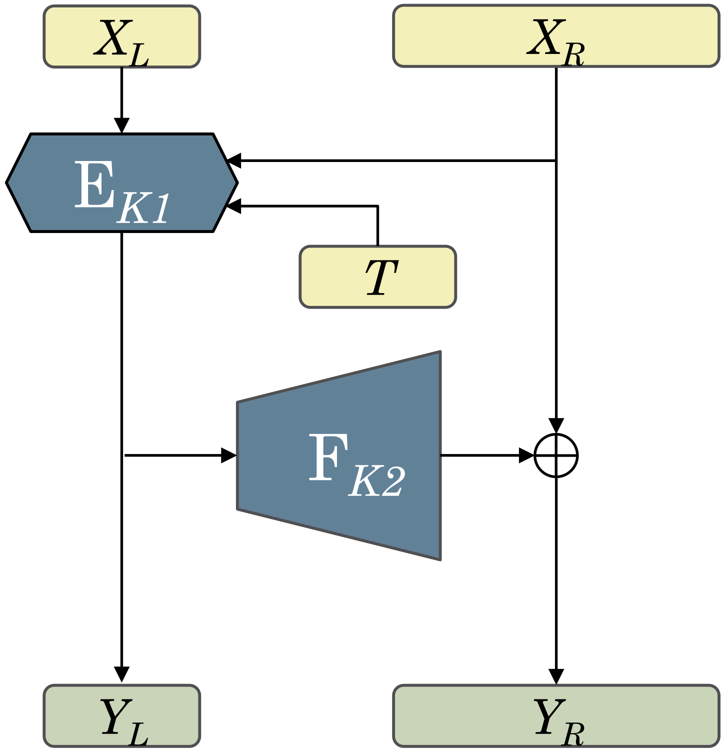
Onion Encryption in Tor with an RPRP



Cryptography Research Centre

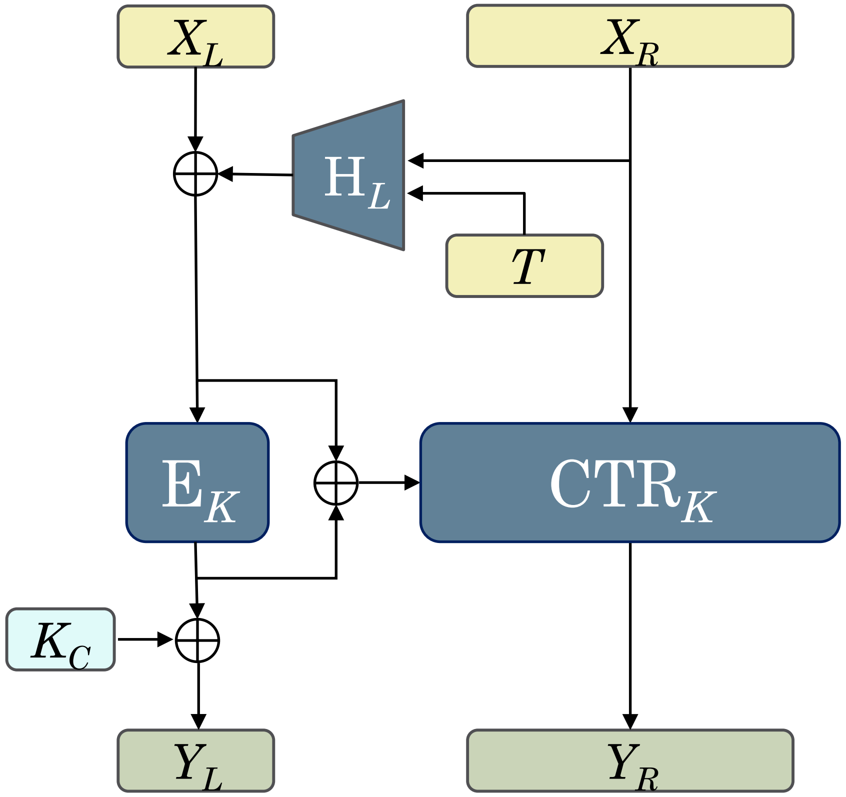
RPRP Constructions

Unilaterally-Protected IV (UIV)



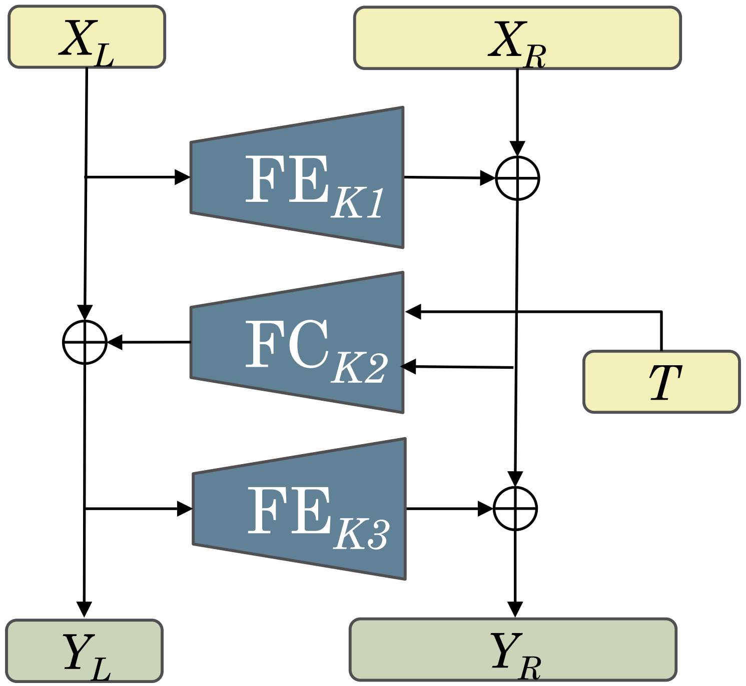
- **UIV** is obtained from **PIV** [ShrTer13] by dropping the **third layer** and is **RPRP secure**.
- It can be instantiated with **GCM components** leading to a **performance** similar to GCM-SIV.
- It is closely related to **GCM-RUP** [ADL17] and **MiniCTR** [Min15].

Hash-Encipher-Counter (HEC)



- HEC is inspired by HCTR where the second AXU Hash is replaced with a lighter XOR operation.
- HEC is an RPRP but not an SPRP.
- When instantiated with GCM components HEC requires less key material than UIV.

Unbalanced Three-Round Feistel



- Security proof typically require either access to Guess or Decipher but not both simultaneously.
- As such it makes sense to consider the notions: **RPRPd** (Enc+Dec) and **RPRPg** (Enc+Gue).
- Then the **Expand-Compress-Expand** (**ECE**) construction shown on the left is **RPRd** secure.
- The analogous **Compress-Expand-Compress (CEC)** construction is **RPRPg** secure.

Concluding Remarks

Summary

- Rugged PRPs strike a **new tradeoff** between **security** and **performance.**
- An RPRP is a rather **versatile primitive** to have in a crypto library as it can easily be turned into **MRAE** (n/nh), **RUPAE** (n/nh), **Nonce-Set AEAD/ Order-Resilient Channels**, **Onion Encryption**.
- We are currently working on RPRP constructions with **BBB security**.
- Intrinsically a variable-length cipher is **not key-committing**. Identifying efficient ways to add this property to our constructions is an interesting open problem.