

# POEx: A Beyond-Birthday-Bound-Secure On-Line Cipher

ArcticCrypt 2016

Christian Forler<sup>1</sup>    Eik List<sup>2</sup>    Stefan Lucks<sup>2</sup>    Jakob Wenzel<sup>2</sup>

<sup>1</sup> Hochschule Schmalkalden, <sup>2</sup> Bauhaus-Universität Weimar  
eik.list (at) uni-weimar.de

18 July 2016

# Agenda

1 Motivation

2 POEx

3 Proof Ideas

4 Instantiation

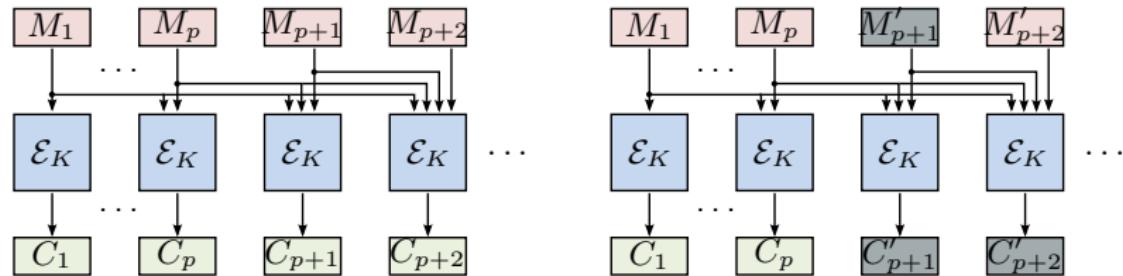
5 Summary

# Section 1

## Motivation

# On-Line Ciphers

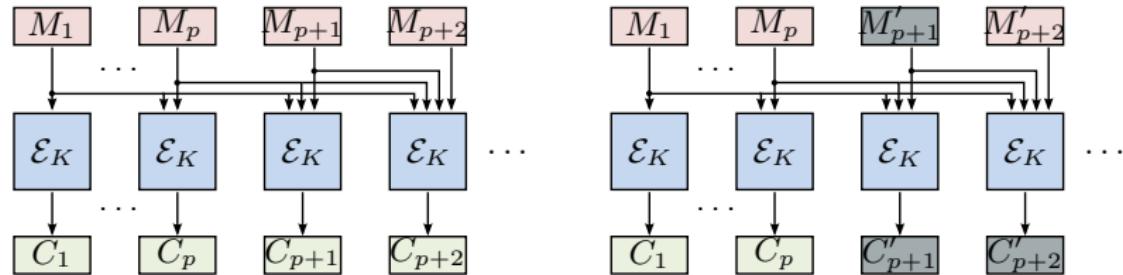
[Bellare et al., 2001]



- On-line cipher:
  - Every  $C_i$  depends only on  $M_1, \dots, M_i$
  - [Boldyreva and Taesombut, 2004]: Constant latency and memory

# On-Line Ciphers

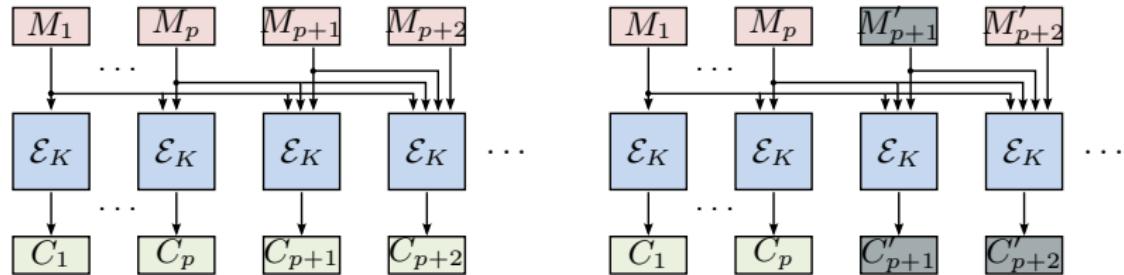
[Bellare et al., 2001]



- On-line cipher:
  - Every  $C_i$  depends only on  $M_1, \dots, M_i$
  - [Boldyreva and Taesombut, 2004]: Constant latency and memory
- Length-preserving

# On-Line Ciphers

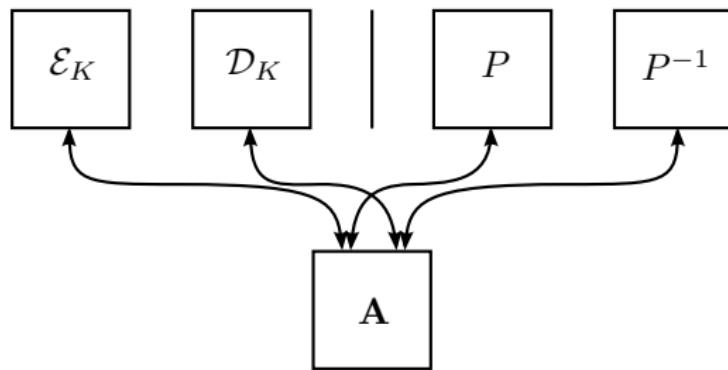
[Bellare et al., 2001]



- On-line cipher:
  - Every  $C_i$  depends only on  $M_1, \dots, M_i$
  - [Boldyreva and Taesombut, 2004]: Constant latency and memory
- Length-preserving
- Prefix-preserving
  - $p \leftarrow \text{LLCP}_n(M, M')$ : Length (in blocks) of longest common prefix
  - $C_i = C'_i$ , for all  $1 \leq i \leq p$
  - $C_{p+1} \neq C'_{p+1}$
  - $C_i, C'_i$  independent for all  $i > p + 1$

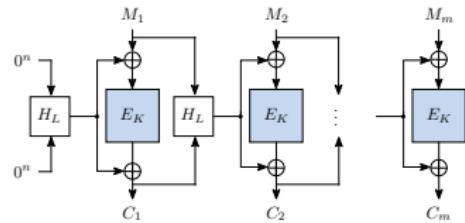
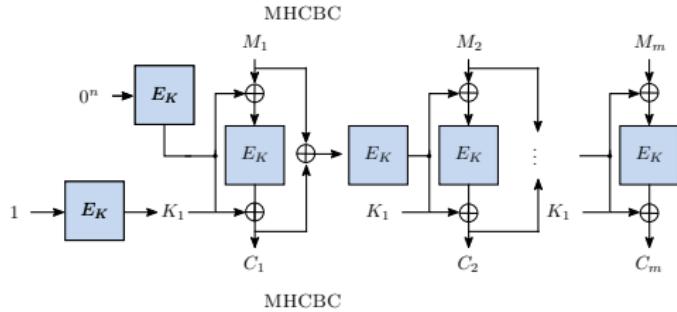
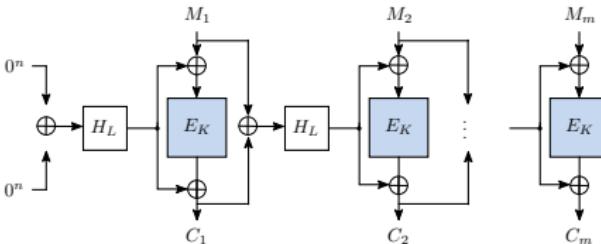
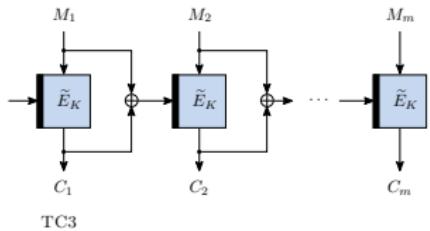
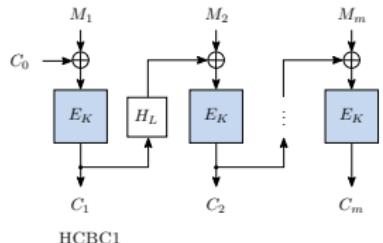
# Notions: SOPRP-Security

[Bellare et al., 2001]



- $P \leftarrow \text{OPerm}_n$
- $K \leftarrow \mathcal{K}$

# Limitation: Birthday Bound



- (S)OPRP security requires dependency of previous block  
     $\implies$  chaining
- All of the above:  $n$ -bit chaining value (bottleneck: collision)
- Birthday bound: security lost after  $2^{n/2}$  blocks encrypted under the same key
- Interesting problem in practice and theory

# Application: On-Line Authenticated Encryption Schemes

## Relevance:

- High-throughput/low-latency requirements,  
e. g. Optical Transport Networks [ITU-T, 2009]
- Stream-oriented interfaces in implementations, e. g.  
EVP\_DecryptUpdate in OpenSSL [Young and Hudson, 2011]
- Output (part of) the result before all input parts are fully processed

# Application: On-Line Authenticated Encryption Schemes

## Relevance:

- High-throughput/low-latency requirements,  
e. g. Optical Transport Networks [ITU-T, 2009]
- Stream-oriented interfaces in implementations, e. g.  
EVP\_DecryptUpdate in OpenSSL [Young and Hudson, 2011]
- Output (part of) the result before all input parts are fully processed

## 2nd-Round BC-Based Robust On-Line CAESAR Candidates:

- AES-JAMBU, COLM (AES-COPA + ELmD), POET, SHELL

# Application: On-Line Authenticated Encryption Schemes

## Relevance:

- High-throughput/low-latency requirements,  
e. g. Optical Transport Networks [ITU-T, 2009]
- Stream-oriented interfaces in implementations, e. g.  
EVP\_DecryptUpdate in OpenSSL [Young and Hudson, 2011]
- Output (part of) the result before all input parts are fully processed

## 2nd-Round BC-Based Robust On-Line CAESAR Candidates:

- AES-JAMBU, COLM (AES-COPA + ELmD), POET, SHELL

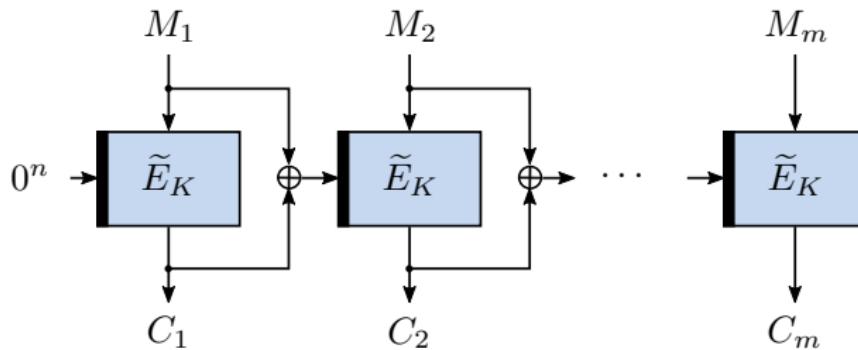
## Inherit birthday-bound limitation

# Approaches for Higher (Provable) Security

- 1 Instantiation with wide-block primitive
- 2 Sponges
- 3 BBB-secure design

# Alternative Approaches

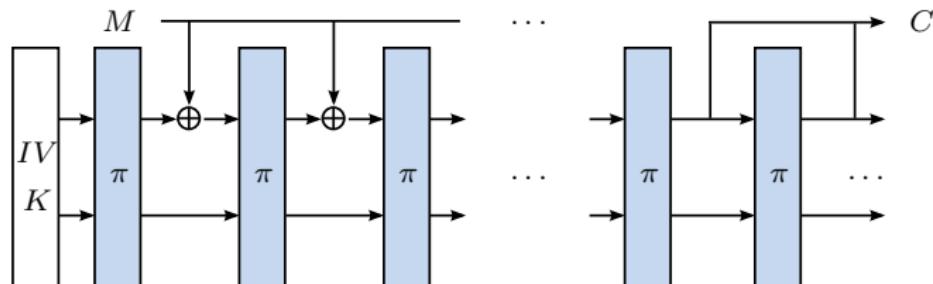
## 1. Instantiation with Wide-Block Primitive



- Example: TC3 [Rogaway and Zhang, 2011] with PRØST permutation or BLAKE2B,  
keyed and tweaked using Even-Mansour [Even and Mansour, 1991]
  - + Efficient
  - + Simple description and analysis
  - Technically not beyond-birthday-bound (BBB)  
(our approach guarantees significantly higher security)

# Alternative Approaches

## 2. Sponge

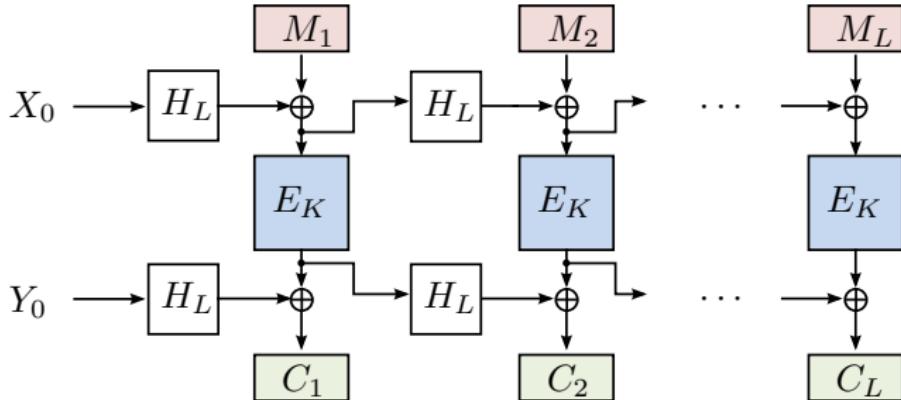


- E.g. Keyak, Ketje, NORX, PRIMATEs, StriBOB, ...
  - + High security margin
  - o Not fully as efficient as block-cipher-based on-line ciphers
  - Technically not BBB

## Section 2

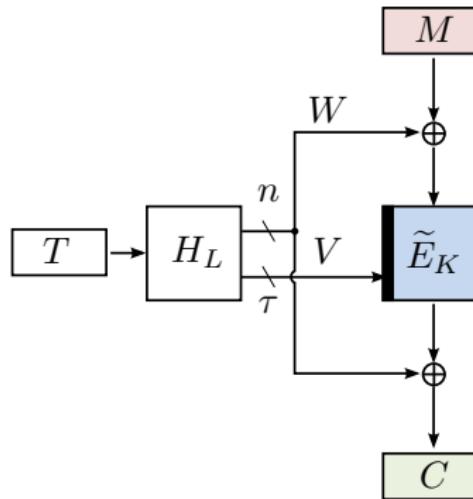
POEx

# POE



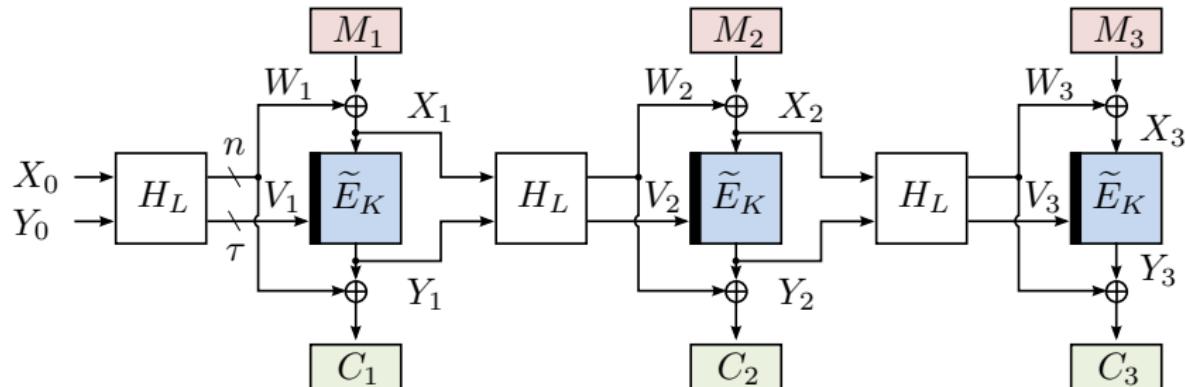
- On-line cipher under POET [Abed et al., 2014]
- 1 BC call + 2 calls to  $\epsilon$ -AXU hash function  $H$  per block
- SOPRP-secure
- POE + PMAC + Tag Splitting:  
Decryption-misuse-resistant on-line AE scheme POET

# XTX



- [Minematsu and Iwata, 2015]
- Tweak-domain extender for tweakable block cipher  
 $\tilde{E} : \mathcal{K} \times \{0, 1\}^\tau \times \{0, 1\}^n \rightarrow \{0, 1\}^n$
- $\epsilon$ -AXU hash function  
 $H : \mathcal{L} \times \{0, 1\}^* \rightarrow \{0, 1\}^\tau \times \{0, 1\}^n$

$$\mathbf{Adv}_{\text{XTX}[\tilde{E}, H], \text{XTX}[\tilde{E}^{-1}, H]^{-1}}^{\text{STPRP}}(\mathbf{A}) \leq \epsilon \cdot q^2 + \mathbf{Adv}_{\tilde{E}, \tilde{E}^{-1}}^{\text{STPRP}}(\ell, O(t)).$$



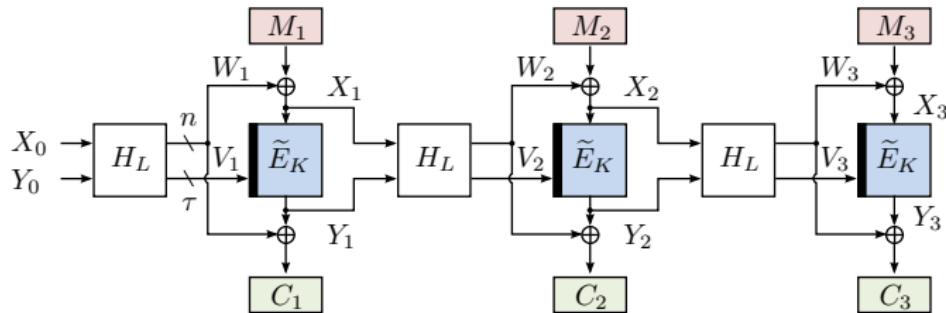
- XTX chained
- $H$ :  $\epsilon$ -AXU hash function
- $\tilde{E}$ : tweakable block cipher
- SOPRP-secure on-line secure up to about  $O(2^{n+\tau/2})$  blocks encrypted under same key
- **BBB-secure**

## Section 3

### Proof Ideas

# Proof Ideas

## Steps

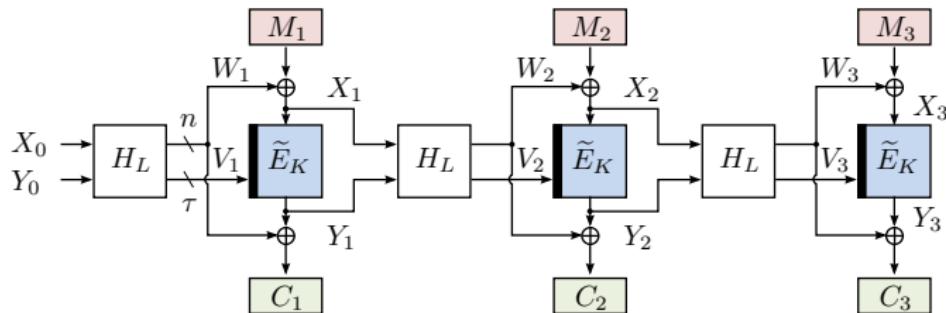


**Steps:**

- 1 Replace  $\tilde{E}$  by ideal primitive  $\tilde{\pi} \leftarrow \text{TPerm}(\tau, n)$

# Proof Ideas

## Steps

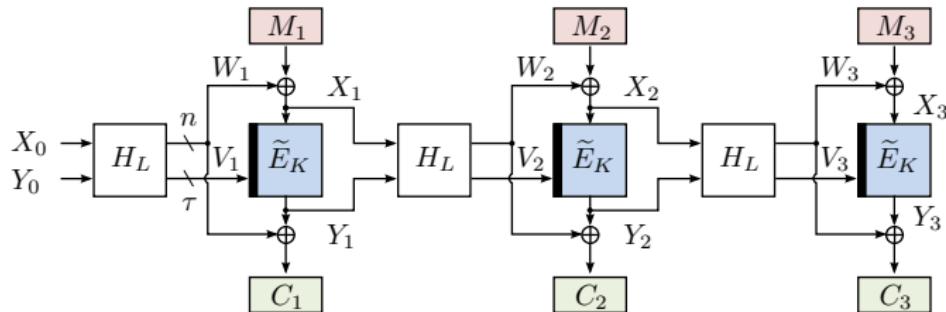


## Steps:

- 1 Replace  $\tilde{E}$  by ideal primitive  $\tilde{\pi} \leftarrow \text{TPerm}(\tau, n)$
- 2 Identify bad events

# Proof Ideas

## Steps

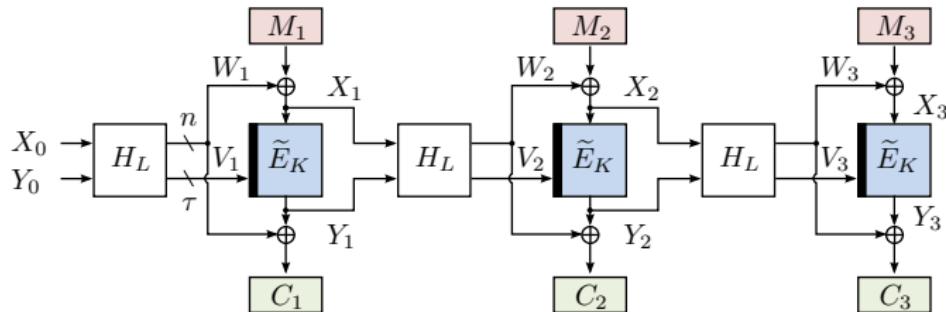


## Steps:

- 1 Replace  $\tilde{E}$  by ideal primitive  $\tilde{\pi} \leftarrow \text{TPerm}(\tau, n)$
- 2 Identify bad events
- 3 Study difference between  $\text{POEx}/\text{POEx}^{-1}$  and  $P/P^{-1}$   
w/o bad events: *In*, *directly after*, and *beyond* common prefix

# Proof Ideas

## Steps

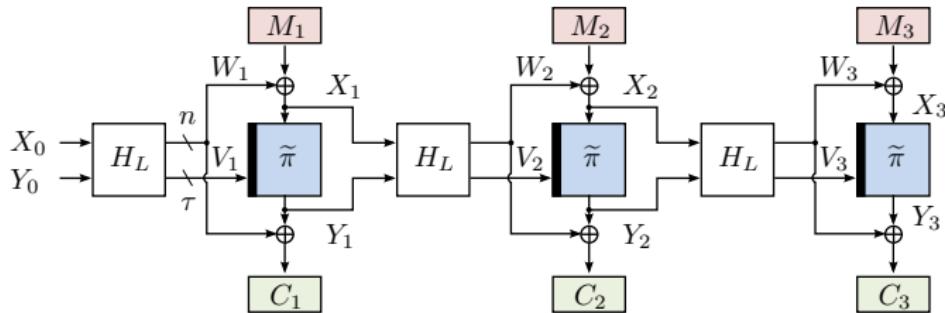


## Steps:

- 1 Replace  $\tilde{E}$  by ideal primitive  $\tilde{\pi} \leftarrow \text{TPerm}(\tau, n)$
- 2 Identify bad events
- 3 Study difference between  $\text{POEx}/\text{POEx}^{-1}$  and  $P/P^{-1}$   
w/o bad events: *In*, *directly after*, and *beyond* common prefix
- 4 Bound probability of bad events

# Proof Ideas

## Bad Events

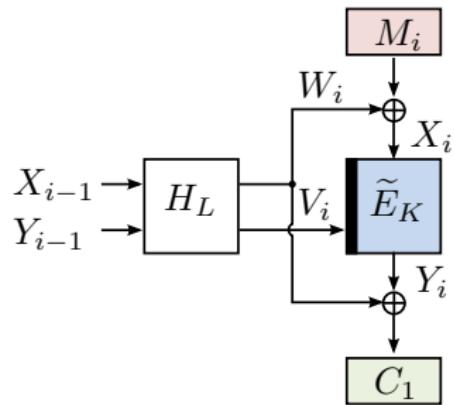


## Bad Events:

- Consider distinct queries:  $(M, C) \neq (M', C')$ ,  $p = \text{LLCP}_n(M, M')$
- Enc. queries: tweak+input collision:  $(V_i, X_i) = (V'_j, X'_j)$
- Enc. queries: chaining-value collision:  $(X_i, Y_i) = (X'_j, Y'_j)$
- Collisions beyond longest common prefix
- Two similar bad events for decryption queries

# Proof Ideas

## Bound



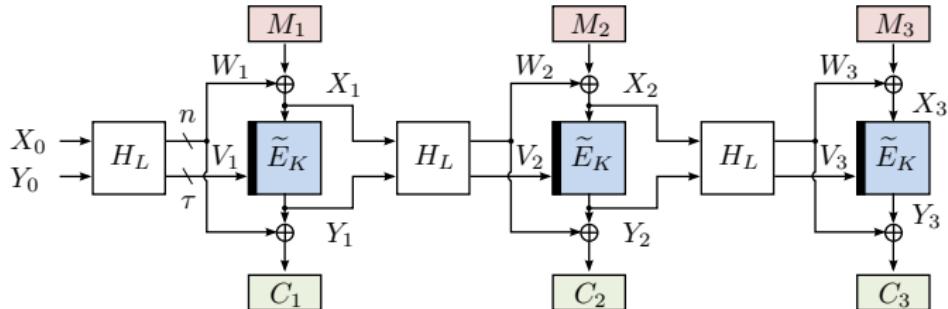
- Assuming independent keys  $K$  and  $L$
- $\epsilon$ -AXU hash function  $H$

$$\mathbf{Adv}_{\text{POEx}[\tilde{E}, H], \text{POEx}[\tilde{E}^{-1}, H]^{-1}}^{\text{SOPRP}}(\mathbf{A}) \leq 2\ell^2\epsilon \cdot \left(2 + \frac{2^\tau}{2^n - \ell}\right) + 2 \cdot \mathbf{Adv}_{\tilde{E}, \tilde{E}^{-1}}^{\text{STPRP}}(\ell, O(t)).$$

## Section 4

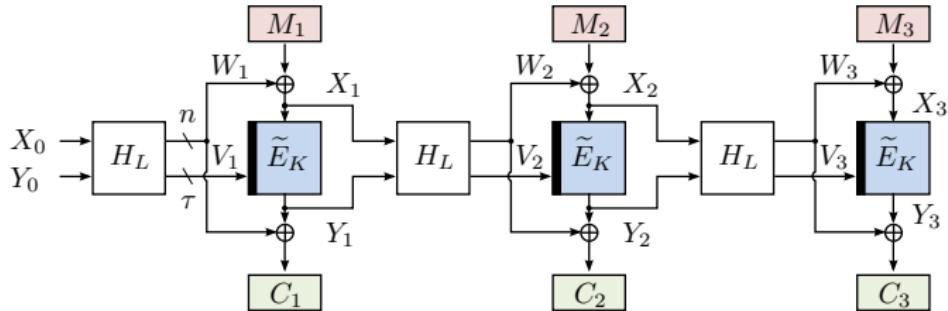
### Instantiation

# Instantiation of $\tilde{E}$



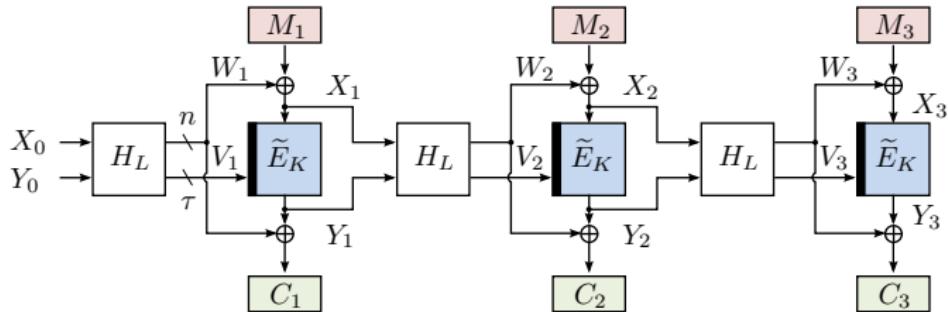
- TWEAKY constructions [Jean et al., 2014]
- Deoxys-BC-128-128 as  $\tilde{E}$ 
  - AES-based, software-efficient
  - 128-bit tweak and state

# Instantiation of $\tilde{E}$



- TWEAK-E constructions [Jean et al., 2014]
- Deoxys-BC-128-128 as  $\tilde{E}$ 
  - AES-based, software-efficient
  - 128-bit tweak and state
- Various application-specific alternatives possible:
  - Joltik-BC, Mennink's designs [Mennink, 2015], ThreeFish [Ferguson et al., 2010], ...

# Instantiation of $H$



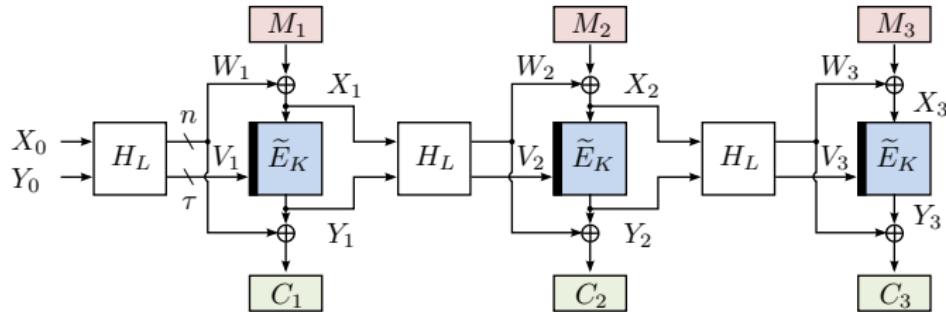
- GF multiplications for  $H$ :

$$\text{POLY}[n]_L(M) := \sum_{i=1}^m L^{m+1-i} \cdot M_i \bmod p_n(x),$$

- $m/2^n$ -AXU for  $\mathbb{GF}(2^n)$ ,  $p_n(x)$ : irreducible polynomial in  $\mathbb{GF}(2^n)$
- For  $\mathcal{L} = \mathbb{GF}(2^n) \times \mathbb{GF}(2^\tau)$ :

$$\text{POLY}[n, \tau]_{L_1, L_2}(M) := (\text{POLY}[n]_{L_1}(M), \text{POLY}[\tau]_{L_2}(M)).$$

# Instantiation of $H$

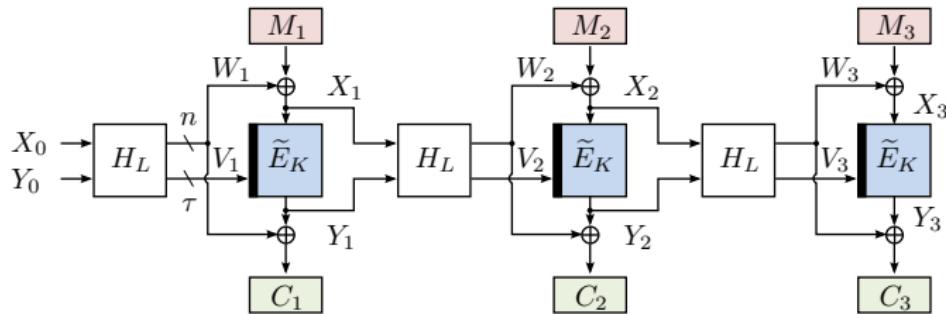


- $\text{POLY}[n, \tau]$  is  $4/2^{n+\tau}$ -AXU for 2-block inputs
- 4 GF multiplications, parallelizable
- For  $\mathcal{L} = \mathbb{GF}(2^n) \times \mathbb{GF}(2^\tau)$  and  $(L_1, L_2) \in \mathcal{L}$ :

$$W_i \leftarrow (L_1^2 \cdot X_{i-1}) + (L_1 \cdot Y_{i-1}) \bmod p_n(x),$$
$$V_i \leftarrow (L_2^2 \cdot X_{i-1}) + (L_2 \cdot Y_{i-1}) \bmod p_\tau(x)$$

where multiplications and additions are defined over  $\mathcal{L}$

# Instantiation



- $\Pi := \text{POEx}[\tilde{E}, \text{POLY}[n, \tau]]$ .
- $\ell$ : #Blocks over all queries
- Assuming  $\ell \leq 2^{n-1}$ :

$$\mathbf{Adv}_{\Pi, \Pi^{-1}}^{\text{SOPRP}}(\mathbf{A}) \leq 16\ell^2 \cdot \left( \frac{1}{2^{n+\tau}} + \frac{1}{2^{2n}} \right) + 2 \cdot \mathbf{Adv}_{\tilde{E}, \tilde{E}^{-1}}^{\text{STPRP}}(\ell, O(t)).$$

## Section 5

### Summary

# Comparison

Aspect	POEx	On-line ciphers										OAE schemes			
		COPE	HCBC1	HCBC2	HPCBC	M CBC	MHCBC	POE	TC1	TC2	TC3	COLM	McOE-G	McOE-X	OLEF
#(T)BC calls	$m$	$2m$	$m$	$m$	$m + 1$	$m$	$m$	$m$	$m$	$m$	$m$	$2m$	$m$	$m$	$2m$
#HF calls	$2m$	—	$m$	$2m$	$2m + 1$	$m$	$2m$	$2m$	—	—	—	—	$m$	—	—
#Keys	2	1	2	2	2	1	2	2	1	1	1	1	2	1	1
HF Key Length	$n + \tau$	—	$n$	$2n$	$2n$	—	$n$	$n$	—	—	—	—	$n$	—	—
SOPRP-secure	•	—	—	•	•	•	•	•	—	•	•	—	—	—	•
BBB	•	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table: • = Provides feature. — = Lacks feature/none.

# Summary

## Features:

- Based on tweakable block cipher + universal hash function
- BBB
- Provably secure if TBC secure

# Summary

## Features:

- Based on tweakable block cipher + universal hash function
- BBB
- Provably secure if TBC secure

## Current Limitations:

- Requires tweakable block cipher + universal hash function
- Pipelinable = sequential calls to TBC
- 2 keys,  $2n$ -bit hash key

# Summary

## Features:

- Based on tweakable block cipher + universal hash function
- BBB
- Provably secure if TBC secure

## Current Limitations:

- Requires tweakable block cipher + universal hash function
- Pipelinable = sequential calls to TBC
- 2 keys,  $2n$ -bit hash key

## Future Work:

- Extend to a BBB-secure on-line AE scheme

Questions?



Lunch?



## Section 6

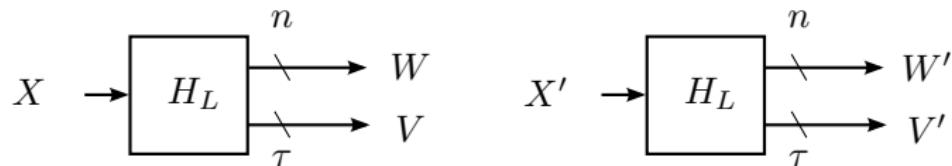
### Supporting Slides

# Bibliography

-  Abed, F., Forler, C., McGrew, D., List, E., Fluhrer, S., Lucks, S., and Wenzel, J. (2014). Pipelineable On-line Encryption.  
In Cid, C. and Rechberger, C., editors, *FSE*, volume 8540 of *Lecture Notes in Computer Science*, pages 205–223. Springer.
-  Bellare, M., Boldyreva, A., Knudsen, L. R., and Namprempre, C. (2001). Online Ciphers and the Hash-CBC Construction.  
In Kilian, J., editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 292–309. Springer.
-  Boldyreva, A. and Taesombut, N. (2004). Online Encryption Schemes: New Security Notions and Constructions.  
In Okamoto, T., editor, *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 1–14. Springer.
-  Even, S. and Mansour, Y. (1991). A Construction of a Cipher From a Single Pseudorandom Permutation.  
In *ASIACRYPT*, pages 210–224.
-  Ferguson, N., Lucks, S., Schneier, B., Whiting, D., Bellare, M., Kohno, T., Callas, J., and Walker, J. (2010). The Skein Hash Function Family.

# AXU/Partial-AXU

[Minematsu and Iwata, 2015]

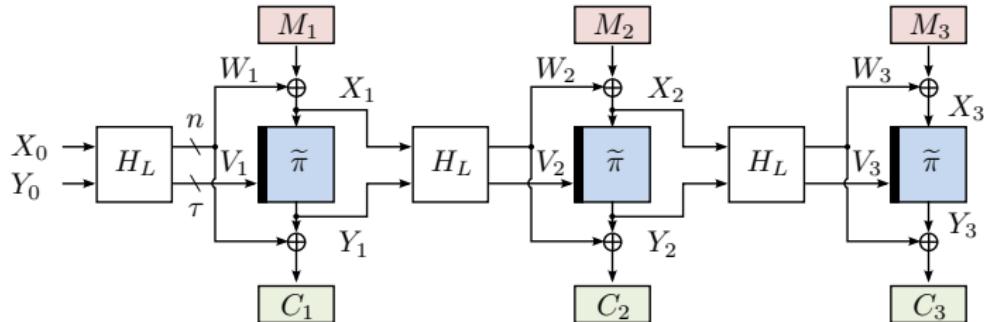


$$\epsilon\text{-AXU} : \max_{\substack{x \neq x' \\ \Delta_1 \in \{0,1\}^{\tau+n}}} \Pr_L [(V \parallel W) \oplus (V' \parallel W') = \Delta_1] \leq \epsilon$$

$$(n, \tau, \epsilon)\text{-pAXU} : \max_{\substack{x \neq x' \\ \Delta_2 \in \{0,1\}^n}} \Pr_L [(V \parallel W) \oplus (V' \parallel W') = (0^\tau \parallel \Delta_2)] \leq \epsilon$$

An  $\epsilon$ -AXU hash function of  $(n + \tau)$ -bit outputs is also  $(n, \tau, \epsilon)$ -pAXU  
[Minematsu and Iwata, 2015]

# Proof Ideas



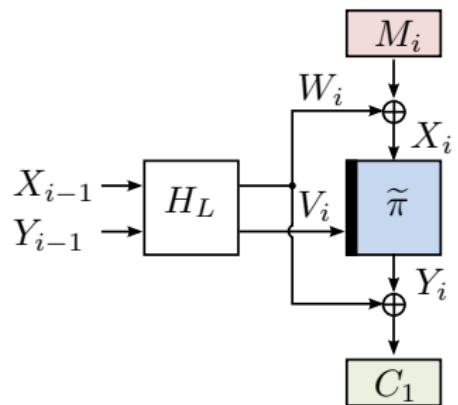
## 1.) Replace $\tilde{E}/\tilde{E}^{-1}$ with Random Tweaked Permutation:

- $\tilde{\pi} \leftarrow \text{TPerm}(\tau, n)$
- Implementable by lazy sampling
- Difference over  $\ell$  blocks

$$\mathbf{Adv}_{\tilde{E}, \tilde{E}^{-1}}^{\text{STPRP}}(\ell, O(t)).$$

# Proof Ideas

## 3.) Behavior without Bad Events

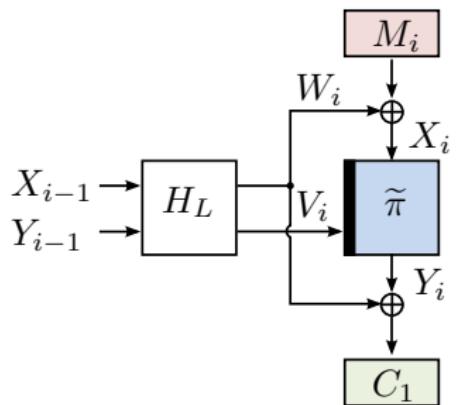


### 3.1) In Common Prefix:

- Same  $(M_i, X_{i-1}, Y_{i-1}) \implies$  same  $C_i$

# Proof Ideas

## 3.) Behavior without Bad Events

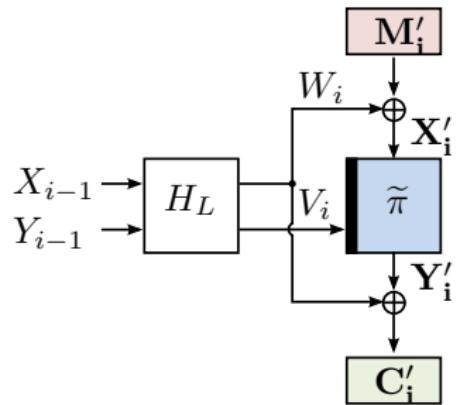


### 3.1) In Common Prefix:

- Same  $(M_i, X_{i-1}, Y_{i-1}) \implies$  same  $C_i$
- Indistinguishable from  $P$

# Proof Ideas

## Behavior without Bad Events

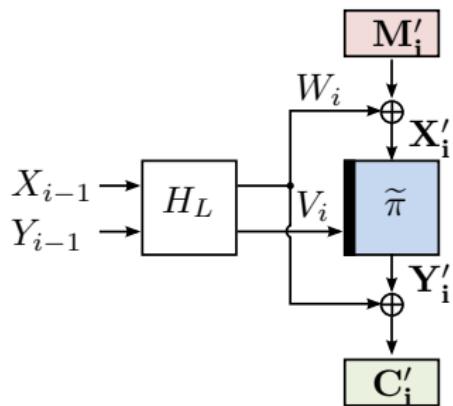


### 3.2) Directly after Common Prefix:

- $(X_{i-1}, Y_{i-1}) = (X'_{i-1}, Y'_{i-1}) \implies (V_{i-1}, W_{i-1}) = (V'_{i-1}, W'_{i-1})$

# Proof Ideas

## Behavior without Bad Events

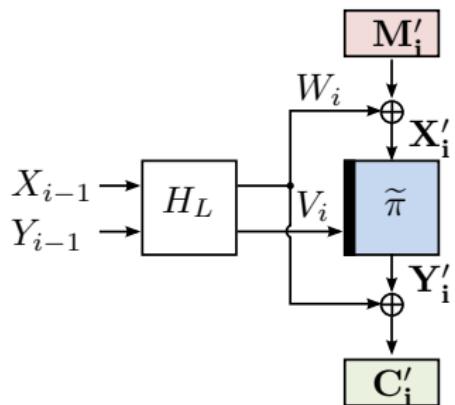


### 3.2) Directly after Common Prefix:

- $(X_{i-1}, Y_{i-1}) = (X'_{i-1}, Y'_{i-1}) \implies (V_{i-1}, W_{i-1}) = (V'_{i-1}, W'_{i-1})$
- $W_i = W'_i$  and  $M_i \neq M'_i \implies X_i \neq X'_i$

# Proof Ideas

## Behavior without Bad Events

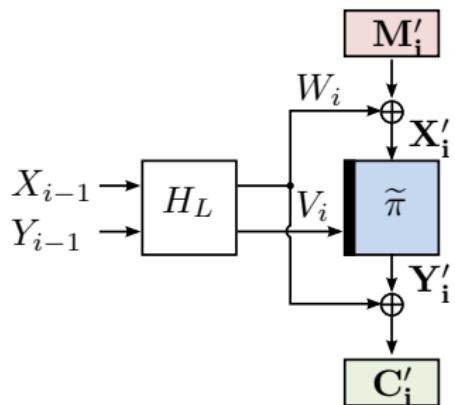


### 3.2) Directly after Common Prefix:

- $(X_{i-1}, Y_{i-1}) = (X'_{i-1}, Y'_{i-1}) \implies (V_{i-1}, W_{i-1}) = (V'_{i-1}, W'_{i-1})$
- $W_i = W'_i$  and  $M_i \neq M'_i \implies X_i \neq X'_i$
- $V_i = V'_i$  and  $X_i \neq X'_i \implies Y_i \neq Y'_i$

# Proof Ideas

## Behavior without Bad Events

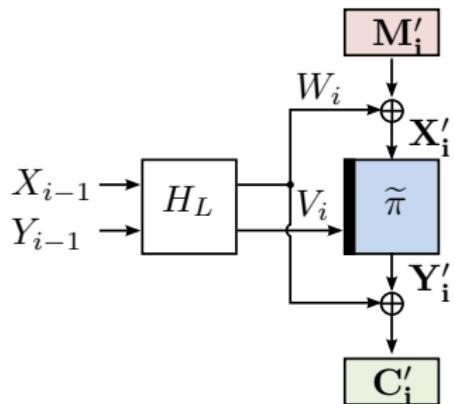


### 3.2) Directly after Common Prefix:

- $(X_{i-1}, Y_{i-1}) = (X'_{i-1}, Y'_{i-1}) \implies (V_{i-1}, W_{i-1}) = (V'_{i-1}, W'_{i-1})$
- $W_i = W'_i$  and  $M_i \neq M'_i \implies X_i \neq X'_i$
- $V_i = V'_i$  and  $X_i \neq X'_i \implies Y_i \neq Y'_i$
- $W_i = W'_i$  and  $Y_i \neq Y'_i \implies C_i \neq C'_i$

# Proof Ideas

## Behavior without Bad Events

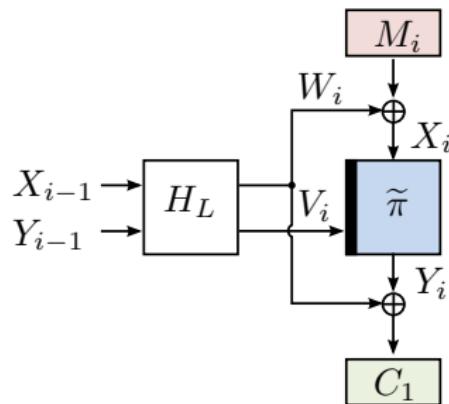


### 3.2) Directly after Common Prefix:

- $(X_{i-1}, Y_{i-1}) = (X'_{i-1}, Y'_{i-1}) \implies (V_{i-1}, W_{i-1}) = (V'_{i-1}, W'_{i-1})$
- $W_i = W'_i$  and  $M_i \neq M'_i \implies X_i \neq X'_i$
- $V_i = V'_i$  and  $X_i \neq X'_i \implies Y_i \neq Y'_i$
- $W_i = W'_i$  and  $Y_i \neq Y'_i \implies C_i \neq C'_i$
- Indistinguishable from  $P$

# Proof Ideas

## Behavior without Bad Events



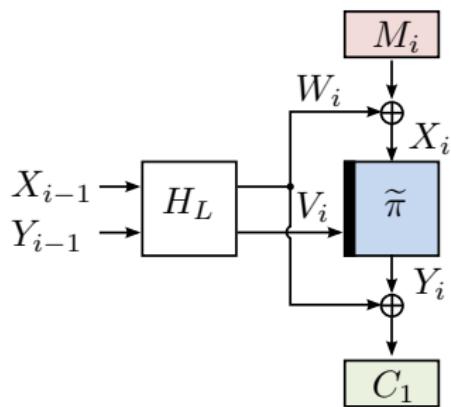
### 3.3) Beyond Common Prefix:

- Assuming no bad events:  $(X_{i-1}, Y_{i-1}, M_i) \neq (X'_{j-1}, Y'_{j-1}, M'_i)$
- Bounded by max. advantage to distinguish  $\text{XTX}[\tilde{\pi}, H]$  from random permutation [Minematsu and Iwata, 2015]

$$\mathbf{Adv}_{\text{XTX}[\tilde{\pi}, H], \text{XTX}[\tilde{\pi}^{-1}, H]^{-1}}^{\text{STPRP}}(\ell, O(t)) \leq \epsilon \cdot \ell^2$$

# Proof Ideas

## 4.) Probability of Bad Events



$$\text{bad}_1 := (V_i = V'_j) \wedge (X_i = X'_j)$$

- Definition of pAXU
- $H$  is  $\epsilon$ -AXU  $\implies H$  is  $\epsilon$ -pAXU
- Over at most  $\ell$  blocks of all queries:

$$\Pr[\text{bad}_1] \leq \epsilon \cdot \ell^2 / 2$$

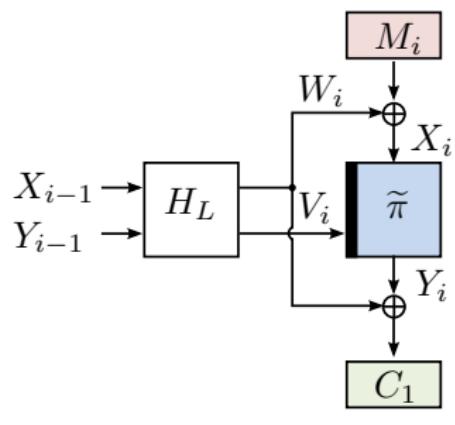
Similar argument in decryption direction:

$$\text{bad}_3 := (V_i = V'_j) \wedge (Y_i = Y'_j)$$

$$\Pr[\text{bad}_3] \leq \text{bad}_1$$

# Proof Ideas

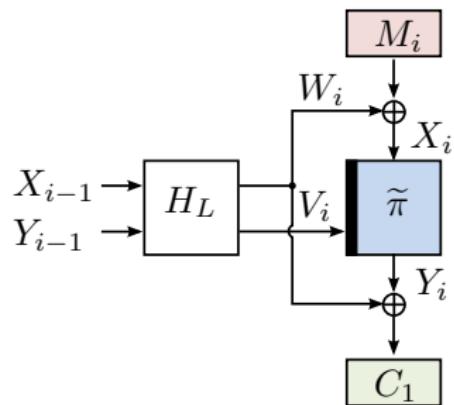
## 4.) Probability of Bad Events



- $\text{bad}_2 := (X_i = X'_j) \wedge (Y_i = Y'_j)$
- $\Pr[\text{bad}_1 \vee \text{bad}_2] \leq \Pr[\text{bad}_1] + \Pr[\neg \text{bad}_1 \wedge \text{bad}_2]$
- $\neg \text{bad}_1 \wedge \text{bad}_2 := \neg[(V_i = V'_j) \wedge (X_i = X'_j)] \wedge [(X_i = X'_j) \wedge (Y_i = Y'_j)]$
- $\neg \text{bad}_1 \wedge \text{bad}_2 := (X_i = X'_j) \wedge (Y_i = Y'_j) \wedge (V_i \neq V'_j)$
- $(V_i \neq V'_j) \implies \text{Independent } \tilde{\pi}^{V_i}, \tilde{\pi}^{V'_j}$

# Proof Ideas

## 4.) Probability of Bad Events



$$\Pr[(X_i = X'_j) \wedge (Y_i = Y'_j) \wedge (V_i \neq V'_j)]$$

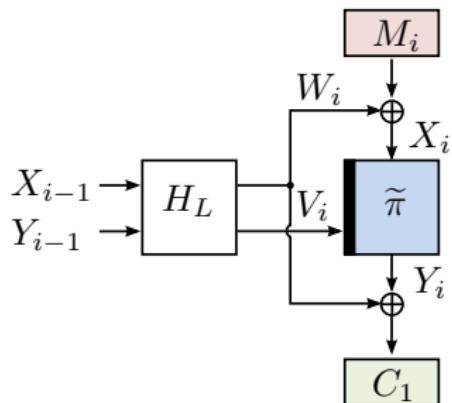
■  $H$  is  $\epsilon$ -pAXU:

$$\begin{aligned}\Pr[X_i = X'_j] &= \Pr[W_i \oplus W'_j = M_i \oplus M'_j] \\ &\leq 2^\tau \cdot \epsilon\end{aligned}$$

since we consider all  $2^\tau - 1$  possible  
 $V_i \neq V'_j$

# Proof Ideas

## 4.) Probability of Bad Events



- Independent  $\tilde{\pi}^{V_i}, \tilde{\pi}^{V'_j}$ :

$$\Pr[Y_i = Y'_j | X_i = X'_j \wedge V_i \neq V'_j] \leq \frac{1}{2^n - \ell}$$

- Over  $\ell$  blocks of all queries:

$$\begin{aligned} &\Pr[Y_i = Y'_j | X_i = X'_j \wedge V_i \neq V'_j] \\ &\quad \cdot \Pr[X_i = X'_j \wedge V_i \neq V'_j] \\ &\leq \frac{\ell^2}{2} \cdot 2^\tau \cdot \epsilon \cdot \frac{1}{2^n - \ell} \end{aligned}$$

- Similar argument in decryption direction