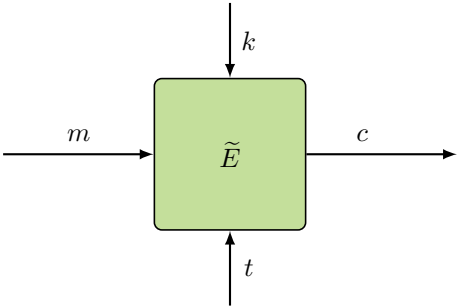
HCTR+: An Optimally Secure TBC-Based Accordion Mode

Nilanjan Datta **Avijit Dutta** **Shibam Ghosh** **Eik List** **Hrithik Nandi**

FSE 2026, Singapore

March 26, 2026

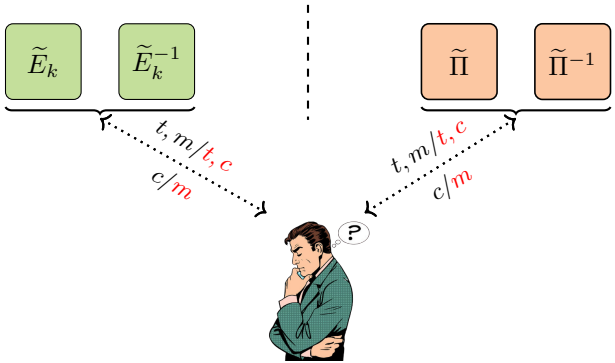
Tweakable Block Cipher



- A family of permutations indexed by secret key and public tweak
- $\widetilde{E}\colon \mathcal{M} \times \mathcal{K} \times \mathcal{T} \to \mathcal{C}$, where $\mathcal{M} \coloneqq \{0,1\}^n, \mathcal{K} \coloneqq \{0,1\}^k$, $\mathcal{T} \coloneqq \{0,1\}^t$ and $\mathcal{C} \coloneqq \{0,1\}^n$
- For a fixed key, tweak pair (k,t), $\widetilde{E}_{k,t}$ is a permutation over $\{0,1\}^n$

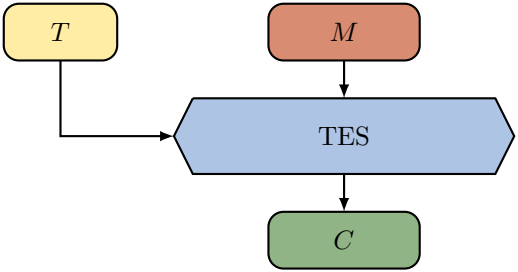
STPRP Security

- $\widetilde{E}$ is a tweakable block cipher and $\widetilde{\Pi}$ is a tweakable random permutation



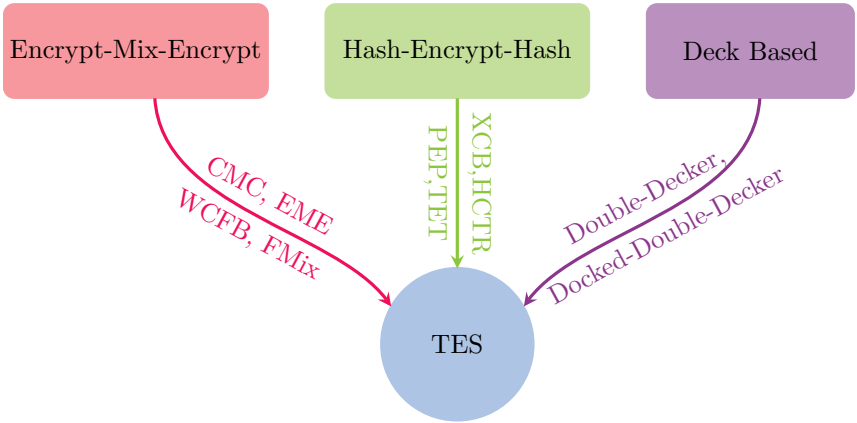
$$\mathrm{Adv}^{\mathrm{STPRP}}_{\mathcal{A},\widetilde{E}}(q) \coloneqq |\Pr[k \xleftarrow{\$} \mathcal{K} \colon \mathcal{A}^{\widetilde{E}_k(\cdot,\cdot),\widetilde{E}_k^{-1}(\cdot,\cdot)} \to 1] - \Pr[\widetilde{\Pi} \xleftarrow{\$} \mathrm{TPerm}(n) \colon \mathcal{A}^{\widetilde{\Pi}(\cdot,\cdot),\widetilde{\Pi}^{-1}(\cdot,\cdot)} \to 1]|$$

Tweakable Enciphering Scheme



- A Tweakable Enciphering Scheme (TES) is a function $\widetilde{\mathcal{E}} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{C}$
- A TES should behave like a variable-length STPRP

The design landscape of TES



Our work is based on Hash-Encrypt-Hash *approach*

Accordion Mode

- Accordion Mode is tweakable variable-input-length strong pseudorandom permutation
- The term accordion signifies that the mode would act as a cipher, not only on a single block but on a range of input sizes and should support arbitrary length tweaks

Existing BBB Secure TES

Construction	Primitive	Tweak	Security (bit)	#Op./block Prim.	$\otimes_n$	$\otimes_{2n}$	#Keys	Message length (bit)
TCT2	BC	✓	$2n/3$	4	–	–	$\ell+\tau+19$	$\geq 2n$
bbb-ddd-AES	BC	✓	$2n/3^{\dagger}$	2	2	–	1	$\geq 2n$
Db-HCTR	BC	✓	$2n/3$	1	4	–	3	$\geq 2n$
AES-CTET^{+}	BC	✓	$2n/3$	2	2	–	6	wn $(w \geq 2)$
DaryaiNoor	BC	✓	n	$\simeq 1$	–	2	1	$\geq 4n$
LargeBlock	TBC	–	n	1	4	–	$\ell+2$	$\geq 2n$
THCTR	TBC	✓	$n^{\dagger}$	1	2	–	3	$\geq 2n$
ZCZ	TBC	–	n	1.5	–	–	1	$\geq 2n$
Kohinoor	TMFC	✓	n	1	–	2	1	$\geq 4n$

ℓ/τ = message length/tweak length and † denotes security degrades gracefully with increasing number of tweak repetitions

Motivation of Our Work

- Beyond the birthday bound Security of an Accordion mode is highly desired
- An accordion mode that accepts variable-length tweaks and provides optimal security can be used to build cryptographic tools that can be used in different applications demanding stronger security
- Until now, almost all of the existing beyond the birthday bound secure tweakable enciphering schemes either fall off from full n-bit security or achieve security that degrades gracefully with tweak repetition

Can we design a TES that provides full n-bit security even in the presence of arbitrary tweak repetition?

Our Contribution: HCTR+ construction

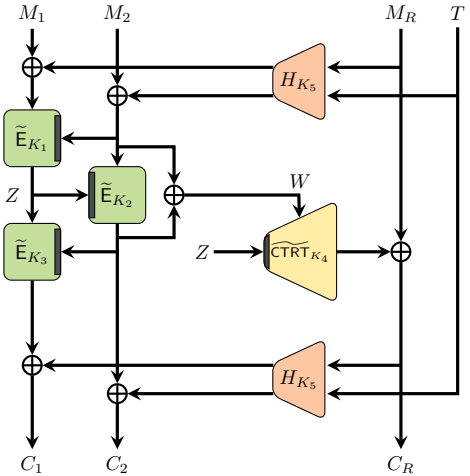


Figure: HCTR+

- H is a $2n$-bit keyed hash function
- $\widetilde{\mathsf{E}}$ is an n-bit tweakable block cipher with n-bit tweak
- $\widetilde{\mathsf{CTRT}}$ is a tweakable block cipher based counter mode encryption
- Five keys are derived from a single master key

A closer look of HCTR+

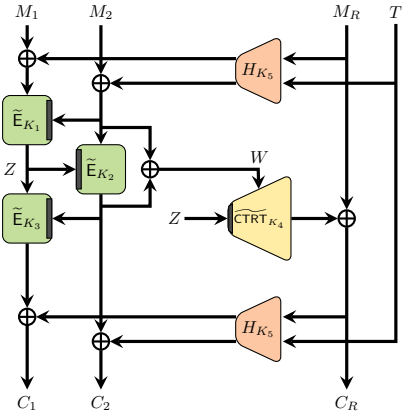


Figure: HCTR+

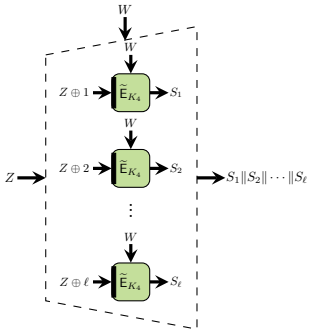


Figure: $\widetilde{\mathsf{CTRT}}_{K_4}$

- In the left part, we use the TLR3 construction, an n-bit secure tweakable SPRP
- In the $\widetilde{\mathsf{CTRT}}$ mode, the tweak of the underlying TBC is incremented in each call

Security Result of HCTR+

Result:

Let $\mathcal{K}$ be a non-empty finite set. Let $\widetilde{\mathsf{E}}$ be an (n, n) tweakable block cipher. Let H be an ϵ-almost-xor-universal $2n$-bit keyed hash function and each $\mathsf{H}[i]$, where $\mathsf{H} = (\mathsf{H}[1], \mathsf{H}[2])$, is an ϵ_i-almost-xor universal n-bit keyed hash function. Then,

$$\mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{HCTR+}[\widetilde{\mathsf{E}},\mathsf{H}]}(\mathsf{A}) \leq 5\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{E}}}(\mathsf{A}') + q^2\epsilon + \frac{q^2\epsilon_2}{2^n} + \frac{2\min\{4q^2\ell_{max}, \sigma^2\}}{2^{2n}} + \frac{3q^2}{2^{2n}} + \frac{10}{2^n},$$

where $q \leq 2^{n-1}$, σ is the total number of message blocks queried, $\ell_{\max}$ is the maximum number of message blocks in any single query.

Proof Overview: Replacing TBCs by random permutations

- Key derivation is replaced by uniform random sampling of five n-bit keys

$$\mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{HCTR+}[\widetilde{\mathsf{E}},\mathsf{H}]}(\mathsf{A}) \leq \mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{\mathsf{E}}}(\mathsf{A}) + \mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{R\text{-}HCTR+}[\widetilde{\mathsf{E}},\mathsf{H}]}(\mathsf{A}) + \frac{10}{2^n}$$

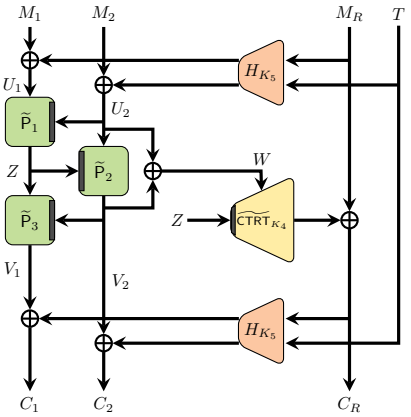
- Replace the keyed tweakable block ciphers with independently sampled tweakable random permutations

$$\mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{R\text{-}HCTR+}[\widetilde{\mathsf{E}},\mathsf{H}]}(\mathsf{A}) \leq 4\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{E}}}(\mathsf{A}) + \underbrace{\mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{R\text{-}HCTR+}^*[\widetilde{\mathsf{P}},\mathsf{H}]}(\mathsf{A})}_{\delta}$$

- Now, our goal is to upper bound δ, where

$$\delta \leq \max_{\mathsf{A}} \left| \Pr[\mathsf{A}^{\mathsf{R\text{-}HCTR+}^*,(\mathsf{R\text{-}HCTR+}^*)^{-1}} = 1] - \Pr[\mathsf{A}^{\mathsf{PP},\mathsf{PP}^{-1}} = 1] \right|$$

Proof Overview: Extended Query Transcript



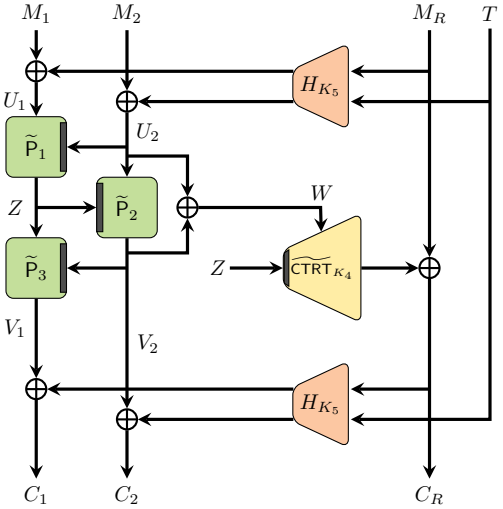
- The transcript generated from the online interaction:

$$\big((T^1, M^1, C^1), \ldots, (T^q, M^q, C^q)\big)$$

- Next, the hash key will be released; in the ideal world a dummy hash key will be sampled
- The intermediate values Z, W will be released; in the ideal world values of W will be computed and values of Z will be sampled maintaining the tweakable permutations property

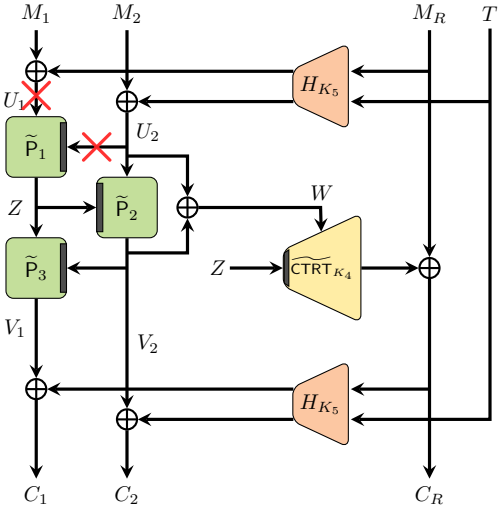
The extended transcript: $\big((T^1, M^1, C^1, Z^1, W^1), \ldots, (T^q, M^q, C^q, Z^q, W^q), K_5\big)$

Proof Overview: Bad Transcripts



Bad events: Non-trivial collisions in the tweak-input or tweak-output pairs

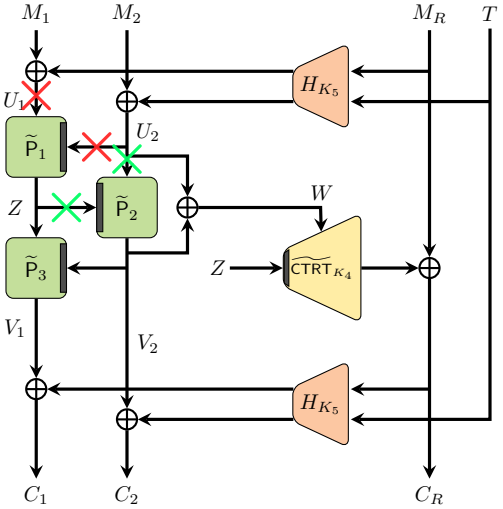
Proof Overview: Bad Transcripts



Bad events: Non-trivial collisions in the tweak-input or tweak-output pairs

- $U_1^i = U_1^j \ \wedge \ U_2^i = U_2^j \ (i \neq j)$

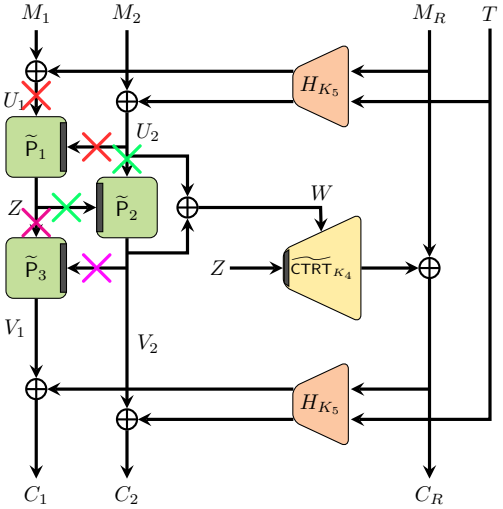
Proof Overview: Bad Transcripts



Bad events: Non-trivial collisions in the tweak-input or tweak-output pairs

- $U_1^i = U_1^j \ \wedge \ U_2^i = U_2^j \ (i \neq j)$
- $Z^i = Z^j \ \wedge \ U_2^i = U_2^j \ (i \in [q_{de}])$

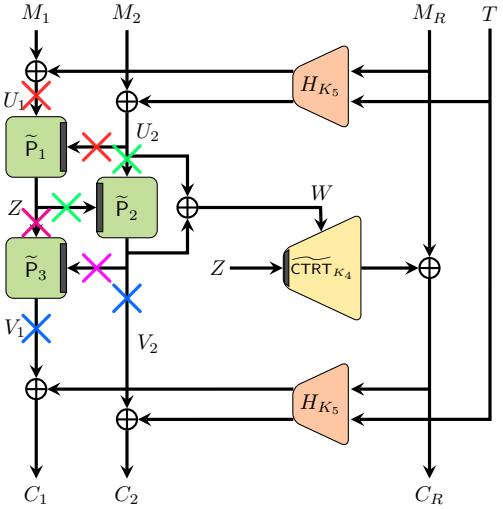
Proof Overview: Bad Transcripts



Bad events: Non-trivial collisions in the tweak-input or tweak-output pairs

- $U_1^i = U_1^j \ \wedge \ U_2^i = U_2^j \ (i \neq j)$
- $Z^i = Z^j \ \wedge \ U_2^i = U_2^j \ (i \in [q_{de}])$
- $Z^i = Z^j \ \wedge \ V_2^i = V_2^j \ (i \in [q_{en}])$

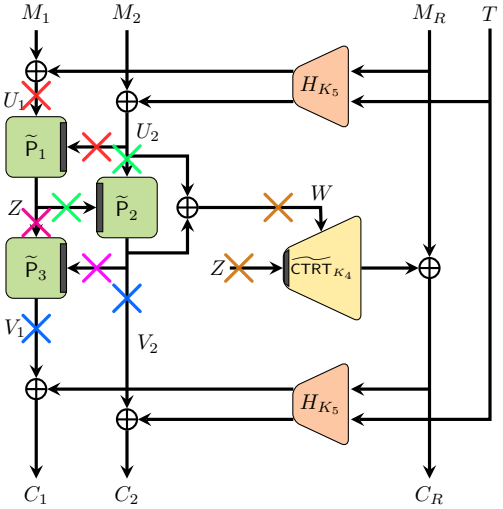
Proof Overview: Bad Transcripts



Bad events: Non-trivial collisions in the tweak-input or tweak-output pairs

- $U_1^i = U_1^j \ \wedge \ U_2^i = U_2^j \ (i \neq j)$
- $Z^i = Z^j \ \wedge \ U_2^i = U_2^j \ (i \in [q_{de}])$
- $Z^i = Z^j \ \wedge \ V_2^i = V_2^j \ (i \in [q_{en}])$
- $V_1^i = V_1^j \ \wedge \ V_2^i = V_2^j \ (i \neq j)$

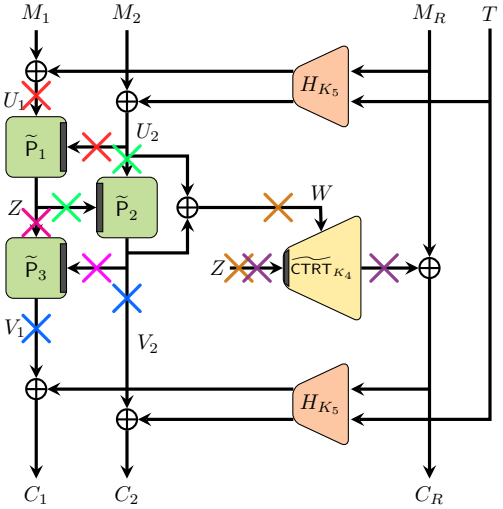
Proof Overview: Bad Transcripts



Bad events: Non-trivial collisions in the tweak-input or tweak-output pairs

- $U_1^i = U_1^j \ \wedge \ U_2^i = U_2^j \ (i \neq j)$
- $Z^i = Z^j \ \wedge \ U_2^i = U_2^j \ (i \in [q_{de}])$
- $Z^i = Z^j \ \wedge \ V_2^i = V_2^j \ (i \in [q_{en}])$
- $V_1^i = V_1^j \ \wedge \ V_2^i = V_2^j \ (i \neq j)$
- $Z^i \oplus \alpha = Z^j \oplus \beta \ \wedge \ W^i = W^j \ (i \neq j)$

Proof Overview: Bad Transcripts



Bad events: Non-trivial collisions in the tweak-input or tweak-output pairs

- $U_1^i = U_1^j \ \wedge \ U_2^i = U_2^j \ (i \neq j)$
- $Z^i = Z^j \ \wedge \ U_2^i = U_2^j \ (i \in [q_{de}])$
- $Z^i = Z^j \ \wedge \ V_2^i = V_2^j \ (i \in [q_{en}])$
- $V_1^i = V_1^j \ \wedge \ V_2^i = V_2^j \ (i \neq j)$
- $Z^i \oplus \alpha = Z^j \oplus \beta \ \wedge \ W^i = W^j \ (i \neq j)$
- $Z^i \oplus \alpha = Z^j \oplus \beta \ \wedge$ $M^i_{\alpha+2} \oplus C^i_{\alpha+2} = M^j_{\beta+2} \oplus C^j_{\beta+2} \ (i \neq j)$

Proof Overview: Bad Transcripts

Let Θ denotes the set of all attainable transcripts τ such that it satisfies one of the above mentioned bad conditions.

Lemma (Bad transcript)

For any integer q such that $q \leq 2^{n-1}$, we have

$$\Pr[\mathsf{X}_{\mathrm{id}} \in \Theta] \leq q^2\epsilon + \frac{q^2\epsilon_2}{2^n} + \frac{2\min\{4q^2\ell_{max}, \sigma^2\}}{2^{2n}} + \frac{2q^2}{2^{2n}},$$

where σ is the total number of message blocks queried, $\ell_{\max}$ is the maximum number of message blocks in any single query.

Proof Overview: Good Transcript Analysis

- Real World: Counted the number of calls of each tweakable random permutation for each distinct tweak
- Ideal World: The probability of the intermediate variables Z have to be calculated

Lemma (Good transcript)

Let $\tau = \left((T^1, M^1, C^1, Z^1, W^1), \ldots, (T^q, M^q, C^q, Z^q, W^q), K_5\right)$ *be a good transcript. Then*

$$\frac{\Pr[\mathsf{X}_{\mathrm{re}} = \tau]}{\Pr[\mathsf{X}_{\mathrm{id}} = \tau]} \geq 1 - \frac{q^2}{2^{2n}}.$$

- Finally, using H-Coefficient technique we derive the advantage bound of the construction

Concrete Instantiation: PHCTR+

- PHCTR+: underlying hash function is PHASH+ and the TBC is Deoxys-BC-128-128 with 128-bit key, tweak, and state size

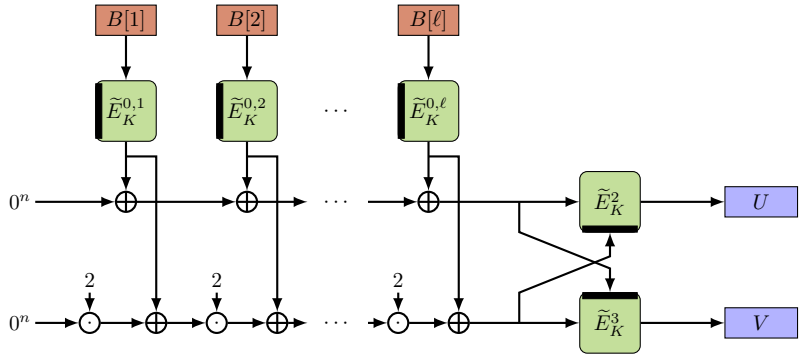


Figure: PHASH+

Concrete Instantiation: ZHCTR+

- ZHCTR+: underlying hash function is ZHASH+ and the TBC is Deoxys-BC-128-128

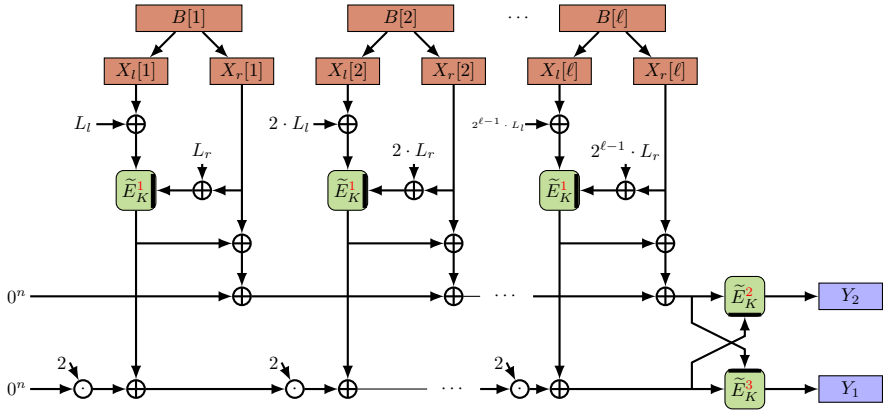


Figure: ZHASH+

Software Performance of ZHCTR+ and PHCTR+

Table: Software performance (c/b) of PHCTR+ and ZHCTR+, and comparison with ZCZ, KohiNoor (KN) and DaryaiNoor (DN). The colors highlight the columns that correspond to the tweak lengths of ZCZ, KohiNoor, or DaryaiNoor.

	ZHCTR+			PHCTR+			ZCZ	KN	DN
Msg./Tweak	0 B	16 B	256 B	0 B	16 B	256 B	0 B	16 B	16 B
1 KB	1.64	1.69	1.41	1.58	1.60	1.35	6.77	1.27	1.47
2 KB	1.32	1.35	1.23	1.38	1.39	1.28	5.97	1.07	1.34
4 KB	1.15	1.17	1.12	1.28	1.29	1.23	5.42	0.98	1.28
8 KB	1.07	1.08	1.05	1.23	1.23	1.20	5.16	0.94	1.25
16 KB	1.03	1.04	1.02	1.20	1.21	1.19	4.84	0.91	1.23
32 KB	1.01	1.01	1.01	1.19	1.19	1.18	4.57	0.91	1.22
64 KB	1.00	1.00	1.00	1.20	1.20	1.19	4.44	0.90	1.22

The source codes of the implementation are available at https://github.com/ShibamCrS/HCTR_PLUS.git.

ZHCTR+: An Attack & A Fix

- Iwata et al. identified a vulnerability in ZHASH+, leading to an attack on ZHCTR+
- Root cause: truncating the first 3 bits after masking with L_r, replaced by a fixed 3-bit value

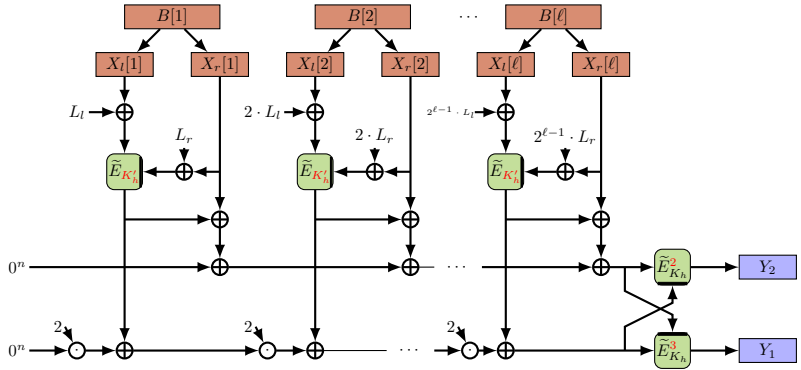


Figure: ZHASH+

- Fix: use the full n-bit masked value as tweak with distinct keys, restoring the security proof

Software Performance after Fixing ZHCTR+

Table: Software performance (c/b) of PHCTR+ and ZHCTR+, and comparison with ZCZ, KohiNoor (KN) and DaryaiNoor (DN). The colors highlight the columns that correspond to the tweak lengths of ZCZ, KohiNoor, or DaryaiNoor.

	ZHCTR+			PHCTR+			ZCZ	KN	DN
Msg./Tweak	0 B	16 B	256 B	0 B	16 B	256 B	0 B	16 B	16 B
1 KB	1.68	1.73	1.45	1.58	1.60	1.35	6.77	1.27	1.47
2 KB	1.36	1.39	1.27	1.38	1.39	1.28	5.97	1.07	1.34
4 KB	1.19	1.21	1.16	1.28	1.29	1.23	5.42	0.98	1.28
8 KB	1.12	1.13	1.10	1.23	1.23	1.20	5.16	0.94	1.25
16 KB	1.08	1.09	1.07	1.20	1.21	1.19	4.84	0.91	1.23
32 KB	1.06	1.06	1.06	1.19	1.19	1.18	4.57	0.91	1.22
64 KB	1.05	1.05	1.05	1.20	1.20	1.19	4.44	0.90	1.22

The source codes of the implementation are available at https://github.com/ShibamCrS/HCTR_PLUS.git.

Current Scenario

Construction	Primitive	Tweak	Security (bit)	#Op./block Prim.	$\otimes_n$	$\otimes_{2n}$	#Keys	Message length (bit)
TCT2	BC	✓	$2n/3$	4	–	–	$\ell+\tau+19$	$\geq 2n$
bbb-ddd-AES	BC	✓	$2n/3^{\dagger}$	2	2	–	1	$\geq 2n$
Db-HCTR	BC	✓	$2n/3$	1	4	–	3	$\geq 2n$
AES-CTET^{+}	BC	✓	$2n/3$	2	2	–	6	wn $(w \geq 2)$
DaryaiNoor	BC	✓	n	$\simeq 1$	–	2	1	$\geq 4n$
LargeBlock	TBC	–	n	1	4	–	$\ell+2$	$\geq 2n$
THCTR	TBC	✓	$n^{\dagger}$	1	2	–	3	$\geq 2n$
ZCZ	TBC	–	n	1.5	–	–	1	$\geq 2n$
Kohinoor	TMFC	✓	n	1	–	2	1	$\geq 4n$
PHCTR+ [**Ours**]	TBC	✓	n	3	–	–	1	$\geq 2n$
ZHCTR+ [**Ours**]	TBC	✓	n	2	–	–	1	$\geq 2n$

ℓ/τ = message length/tweak length and † denotes security degrades gracefully with increasing number of tweak repetitions

Open Problems

- Analyze the multi-user security of HCTR+
- Design a context-committing AE scheme based on HCTR+
- Construct a fully block-cipher-based Accordion mode achieving optimal security

Thank You!

Questions?