Optimally Secure TBC Based Accordion Mode

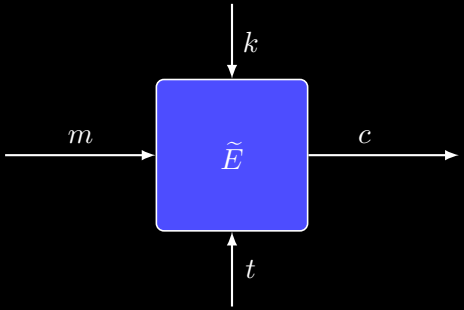
Nilanjan Datta[1] **Avijit Dutta**[1] **Shibam Ghosh**[2] **Hrithik Nandi**[1,3]

[1]Institute for Advancing Intelligence, TCG CREST
[2]University of Haifa, Haifa, Israel
[3]Ramakrishna Mission Vivekananda Educational and Research Institute

The 11th Asian-workshop on Symmetric Key Cryptography (ASK 2024)

December 16, 2024

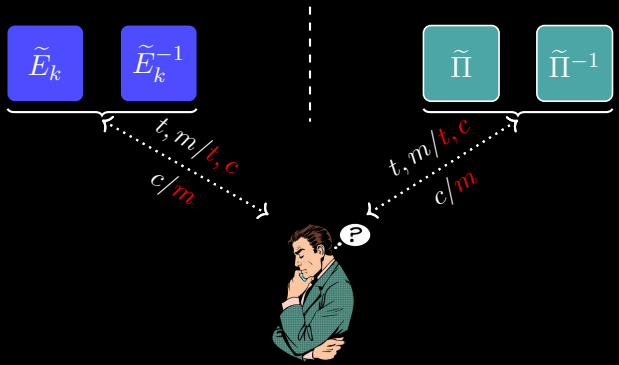
Tweakable Block Cipher



- 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 $\{0,1\}^n$
- TBCs have found diverse applications in designing of AE schemes, MACs, PRFs, Wide block encryption modes

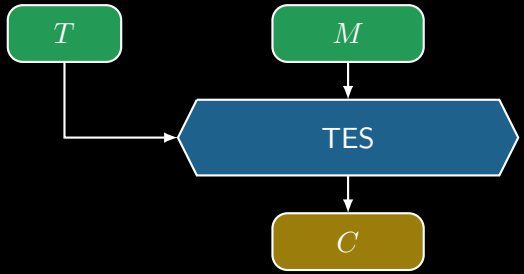
STPRP Security

- $\widetilde{E}$ is a tweakable block cipher and $\widetilde{\Pi}$ is a tweakable random permutation, a family of independent random permutations parameterized by tweak t



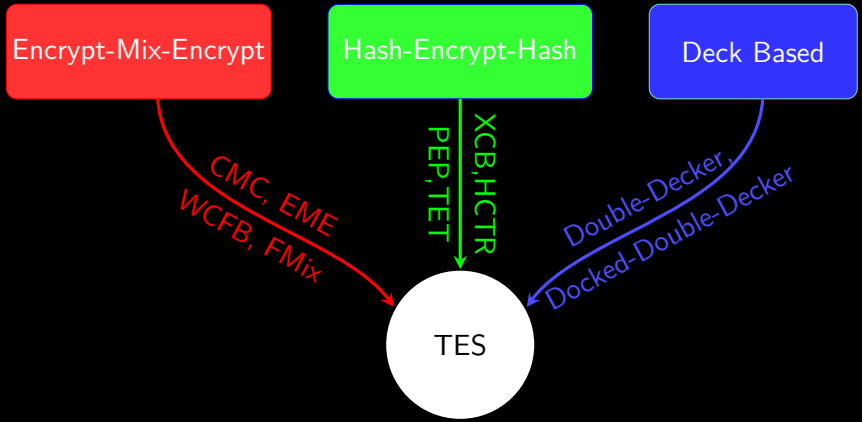
$$\mathsf{Adv}^{\mathsf{STPRP}}_{\mathcal{A},\widetilde{E}}(q) \coloneqq |\Pr[k \xleftarrow{\$} \mathcal{K} \colon \mathcal{A}^{\widetilde{E}_k(\cdot,\cdot),\widetilde{E}_k^{-1}(\cdot,\cdot)} \to 1] - \Pr[\widetilde{\Pi} \xleftarrow{\$} \mathsf{TPerm}(n) \colon \mathcal{A}^{\widetilde{\Pi}(\cdot,\cdot),\widetilde{\Pi}^{-1}(\cdot,\cdot)}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

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 strong tweakable pseudorandom permutation

The design landscape of TES



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	Security	# Ops per Block	# of Keys	Tweak
LargeBlock	TBC	n	1 TBC + 4 FM	$\ell + 2$	$\times$
TCT_2	BC	$2n/3$	4 TBC	$\ell + \tau + 19$	✓
THCTR	TBC	n †	1 TBC + 2 FM	3	✓
bbb-ddd-AES	BC	$2n/3$ †	2 BC + 2 FM	3	✓
Db-HCTR	BC	$2n/3$	1 BC + 4 FM	3	✓
ZCZ	TBC	n	1.5 TBC	1	$\times$

Table: Comparative Study of Beyond the Birthday Bound Secure Tweakable Enciphering Scheme. TBC denotes Tweakable Block Cipher, BC denotes Block Cipher, and FM denotes Field Multiplication. We use ℓ and τ to denote the message and tweak length respectively (in blocks). The † symbol indicates that the security degrades gracefully with maximum repetition of tweaks. Security is mentioned in terms of the number of bits.

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
- Till now, 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

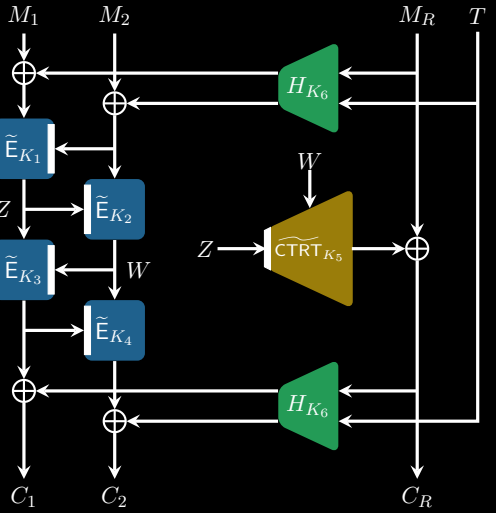


Figure: HCTR+ construction.

- 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, where the tweak of the underlying TBC is incremented in every call of the primitive.

Design Rationale of HCTR+

- The motivation of our construction is to achieve n-bit tweakable sprp security which does not depend on repetition of the tweak.
- We have used TLR4 on the left side and a TBC based counter mode of encryption $\widetilde{\text{CTRT}}$ on the right side of our construction.
- Although, TLR3 ensures optimal security, we have observed that it does not suffice in our context as we need to generate $2n$ bit entropy from the left side of the construction, which will be used in the $\widetilde{\text{CTRT}}$ construction. Otherwise, we would have ended up with security that depends on the tweak repetition.
- In $\widetilde{\text{CTRT}}$ mode, the counter has been used in the tweak part of the underlying TBC instead of incrementing the input value W at every call of the TBC. This role swap has arisen because of a single query distinguishing attack on the THCTR construction in two independent works by Andreeva et al. and Khairallah.

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,

$$\mathbf{Adv}^{\mathrm{STPRP}}_{\mathsf{HCTR}+[\widetilde{\mathsf{E}},\mathsf{H}]}(\mathsf{A}) \leq 6\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{\mathsf{E}}}(\mathsf{A}') + \frac{q^2\epsilon_1}{2^{n+1}} + \frac{q^2\epsilon_2}{2^{n+1}} + q^2\epsilon + \frac{2q^2\ell_{\max}}{2^{2n}} + \frac{2q^2}{2^{2n}} + \frac{15}{2^n},$$

where $\ell_{\max}$ is the maximum number of message blocks queried.

Concrete Instantiation: PHCTR+

We have instantiated the underlying hash function with PHASH+ and the underlying TBC by Deoxys-BC-128-128 with 128-bit key, tweak, and state size.

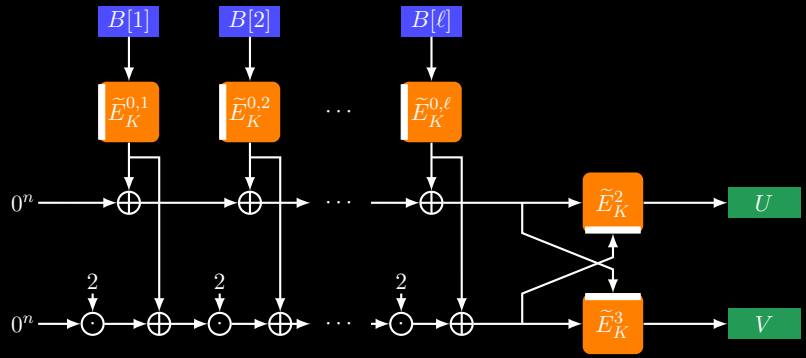


Figure: PHASH+

Concrete Instantiation: ZHCTR+

Instantiating the underlying hash function with ZHASH+ and the TBC by Deoxys-BC-128-128.

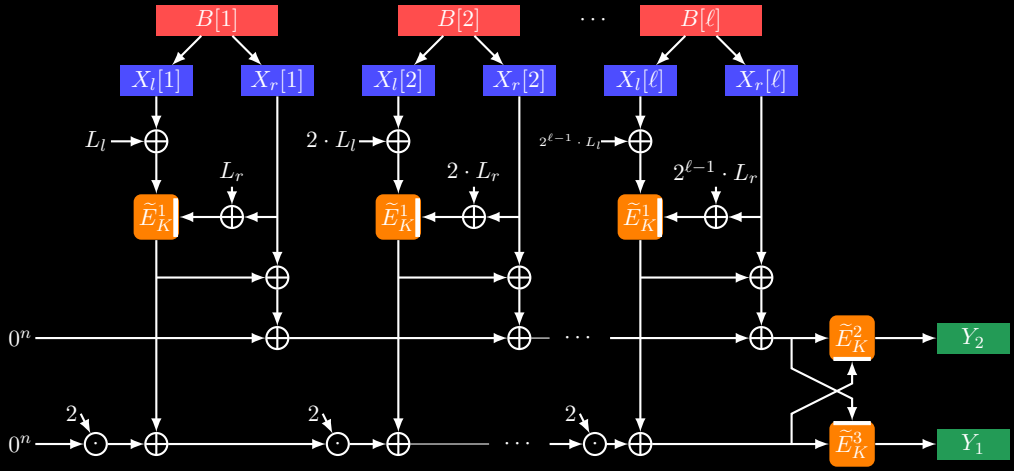


Figure: ZHASH+

Software Performance of ZHCTR+ and PHCTR+

Table: Software performance (CPB) of PHCTR+ and ZHCTR+, and comparison with ZCZ. Note that ZCZ does not accept any tweak, so the comparison of ZCZ should be with zero length tweak of PHCTR+ and ZHCTR+ (red colored columns).

Name	PHCTR+						ZHCTR+						ZCZ
	Tweak Length (Bytes)												
Message Length (Bytes)	0B	16B	32B	64B	128B	256B	0B	16B	32B	64B	128B	256B	-
128B	4.87	4.63	4.20	3.55	2.84	2.10	4.85	4.64	4.10	2.56	2.70	1.97	14.10
256B	3.35	3.32	3.14	2.89	2.50	2.04	3.17	3.17	2.94	2.72	2.32	1.87	11.14
512B	2.62	2.61	2.55	2.42	2.24	1.97	2.29	2.27	2.22	2.14	1.94	1.70	8.31
1KB	2.25	2.24	2.22	2.17	2.08	1.93	1.86	1.84	1.83	1.80	1.70	1.59	6.77
2KB	2.06	2.05	2.04	2.03	1.98	1.91	1.65	1.65	1.63	1.62	1.58	1.52	5.97
4KB	1.97	1.97	1.96	1.95	1.93	1.89	1.54	1.54	1.54	1.53	1.51	1.48	5.42
8KB	1.92	1.92	1.92	1.92	1.91	1.90	1.49	1.49	1.48	1.48	1.47	1.45	5.16
16KB	1.90	1.90	1.90	1.90	1.89	1.88	1.46	1.46	1.46	1.46	1.46	1.45	4.84
32KB	1.89	1.89	1.89	1.89	1.88	1.88	1.45	1.45	1.45	1.45	1.45	1.44	4.57
64KB	1.88	1.88	1.88	1.88	1.88	1.87	1.44	1.44	1.44	1.44	1.44	1.44	4.44

Thank You!