LUND UNIVERSITY

Fast Parallelizable Misuse-Resistant AEAD

LOW LATENCY (DECRYPTION-FAST) SIV

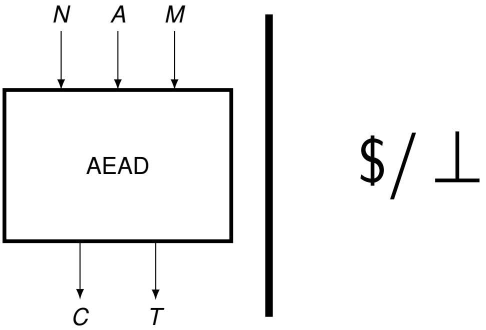
Mustafa Khairallah

LUND UNIVERSITY

Outline

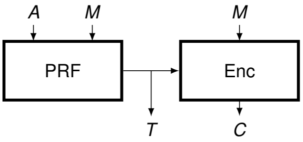
- Why do we care about MRAE?
- Bottleneck of MRAE from TBCs ⟶ Hash-then-PRF MACs.
- Proposal 1: LLSIV.
- Proposal 2: pLLSIV.
- Proposal 3: LLDFV.

AEAD

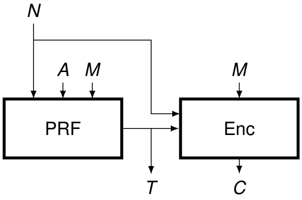


- Online AE: N cannot be repeated.
- DAE: $N =$ constant (or part of A with no restriction).
- MRAE: N can be repeated a (possibly small) number of times.

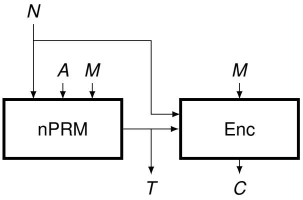
MRAE Design (non-EtE-based)



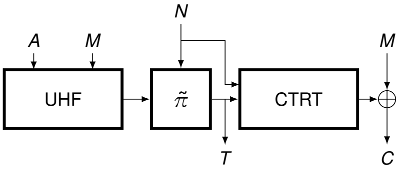
SIV



nSIV

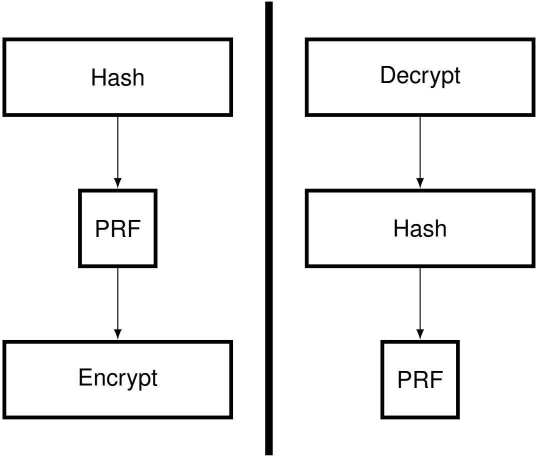


nSIV with nPRM

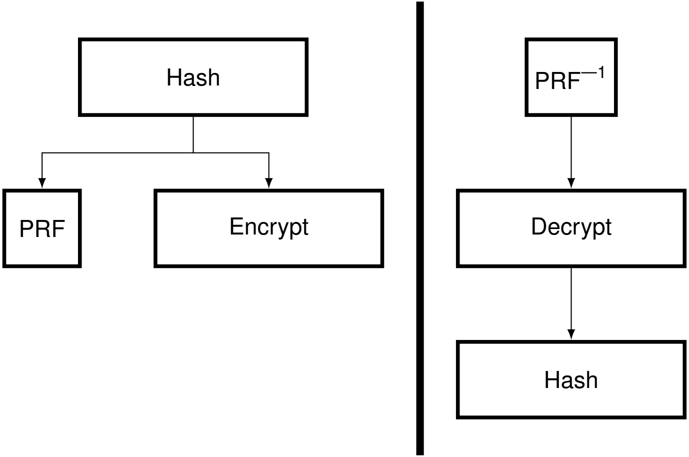


SCT2

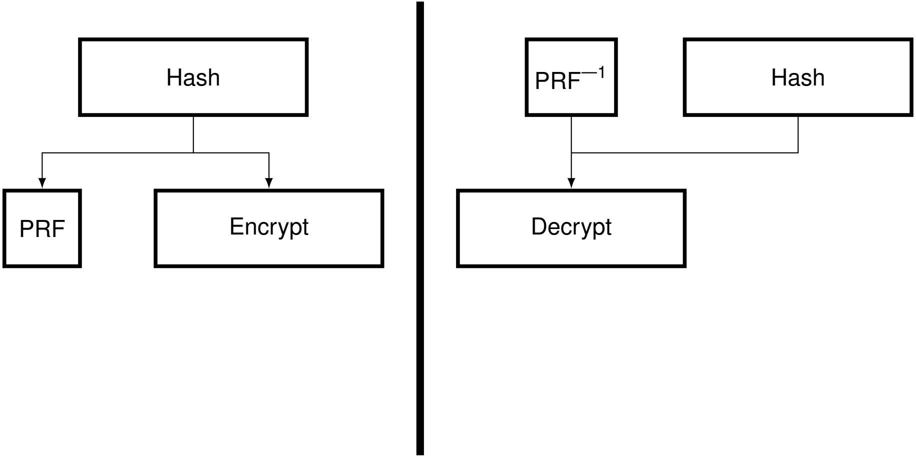
Hash-then-PRF MACs then Encrypt



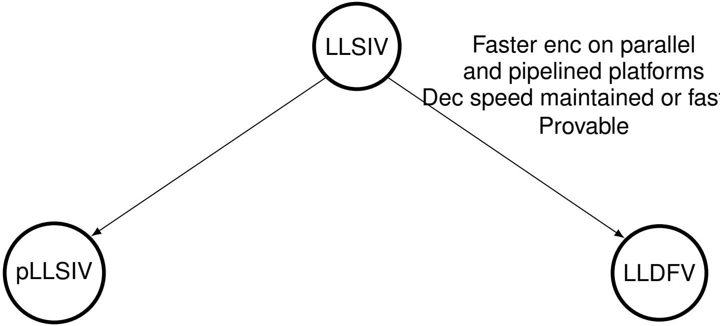
Hash-then-PRF MACs and Encrypt



Hash-then-PRF MACs and Encrypt 2



Proposed Solutions



LLSIV: Faster enc on parallel and pipelined platforms
Dec speed maintained or faster
Provable

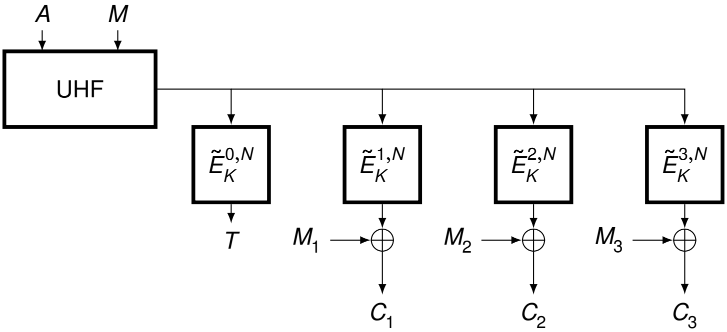
pLLSIV: Faster enc and dec on all platforms
Adhoc/pruned

LLDFV: Faster enc and dec on all platforms
Rate-1 decryption
Provable

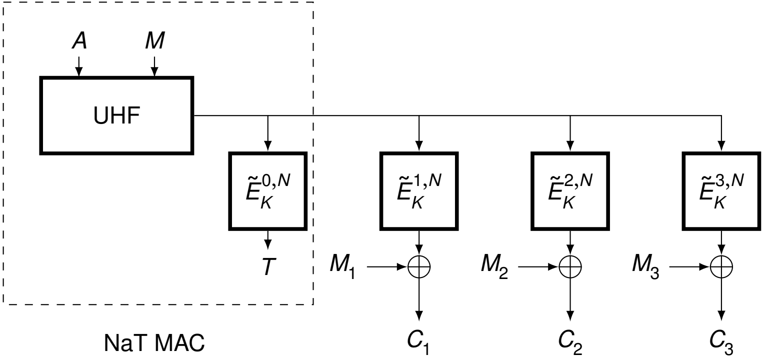
Why optimize encryption and not decryption?

- Encryption is easier to optimize ⟶ more freedom in the data flow.
- Decryption speed is not reduced by this optimization (yet footprint is increased).
- pLLSIV optimizes decryption using adhoc arguments.
- LLDFV optimizes decryption by being an optimization of DFV (later).

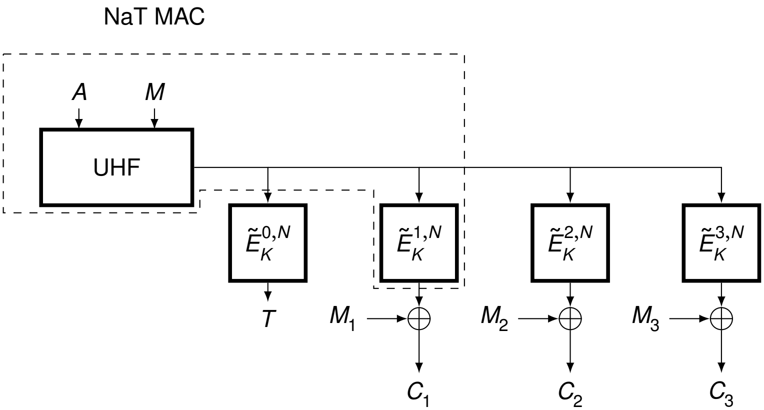
LLSIV


A
M
UHF
$\tilde{E}_K^{0,N}$
$\tilde{E}_K^{1,N}$
$\tilde{E}_K^{2,N}$
$\tilde{E}_K^{3,N}$
T
M_1
M_2
M_3
C_1
C_2
C_3

LLSIV


A
M
UHF
$\tilde{E}_K^{0,N}$
$\tilde{E}_K^{1,N}$
$\tilde{E}_K^{2,N}$
$\tilde{E}_K^{3,N}$
T
M_1
M_2
M_3
C_1
C_2
C_3
NaT MAC

LLSIV



NaT MAC
A
M
UHF
$\tilde{E}_K^{0,N}$
$\tilde{E}_K^{1,N}$
$\tilde{E}_K^{2,N}$
$\tilde{E}_K^{3,N}$
T
M_1
M_2
M_3
C_1
C_2
C_3

Privacy

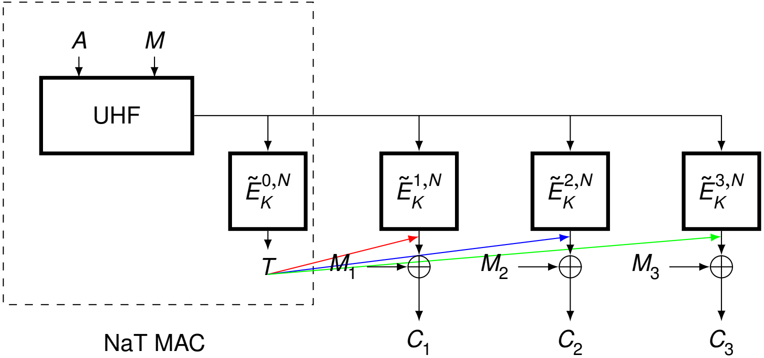
Theorem

Let **A** *be an NM privacy adversary against LLSIV that can repeat a nonce at most* μ *times in encryption queries.* **A** *makes* q_e *queries of total ciphertext size* σ_e *blocks. Let* **A** *run in time at most* t*. Then, there exists a* $(q_e + \sigma_e, t + O(q_e t_H + \sigma_e))$*-TPRP adversary* $\mathbf{A}'$ *against the underlying TBC such that*

$$\mathrm{Adv}_{\mathrm{LLSIV}}^{\mathrm{nm-priv}}(\mathbf{A}) \leq \mathrm{Adv}_{\tilde{E}}^{\mathrm{tprp}}(\mathbf{A}') + (\mu - 1) q_e \epsilon + \frac{(\mu - 1)(q_e + \sigma_e)}{2^n}$$

The hash function $\mathrm{UHF} : \mathcal{K}_h \times \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^n$ *is an* ϵ*-AU hash function and runs in time at most* t_H*.*

LLSIV


A
M
UHF
$\tilde{E}_K^{0,N}$
$\tilde{E}_K^{1,N}$
$\tilde{E}_K^{2,N}$
$\tilde{E}_K^{3,N}$
T
M_1
M_2
M_3
C_1
C_2
C_3
NaT MAC

Authenticity

Theorem

Let **B** *be an NM authenticity adversary against LLSIV that can repeat a nonce at most* μ *times in encryption queries.* **B** *makes* q_e *queries of total ciphertext size* σ_e *blocks and* q_d *decryption/verification queries of total ciphertext size* σ_d*. Let* **B** *run in time at most* t_b*. Then, there exists a* $(q_e + q_d + \sigma_e + \sigma_d, t_b + O((q_e + q_d)t_H + \sigma_e + \sigma_d))$*-sTPRP adversary* $\mathbf{B}'$ *against the underlying TBC such that*

$$\mathrm{Adv}^{\mathrm{nm-auth}}_{\mathrm{LLSIV}}(\mathbf{B}) \le \mathrm{Adv}^{\mathrm{stprp}}_{\tilde{E}}(\mathbf{B}') + \mathrm{Adv}^{\mathrm{mac}}_{\mathrm{NaT}}(\mathbf{B}'')$$

$$\le \mathrm{Adv}^{\mathrm{stprp}}_{\tilde{E}}(\mathbf{B}') + 2(\mu - 1)q_e\epsilon + \frac{q_d}{2^n - \mu} + \mu q_d \epsilon.$$

The hash function $\mathrm{UHF} : \mathcal{K}_h \times \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^n$ *is an* ϵ*-AU hash function and runs in time at most* t_H*.*

Three main steps of the proof

1. Fallback on an SCT-2-like design.
2. Replace remove the depdency between the PRF and the stream cipher.

Lemma

Consider a TBC $\tilde{E} : \mathscr{K} \times \mathscr{I} \times \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$*. Consider the construction* Γ*:*

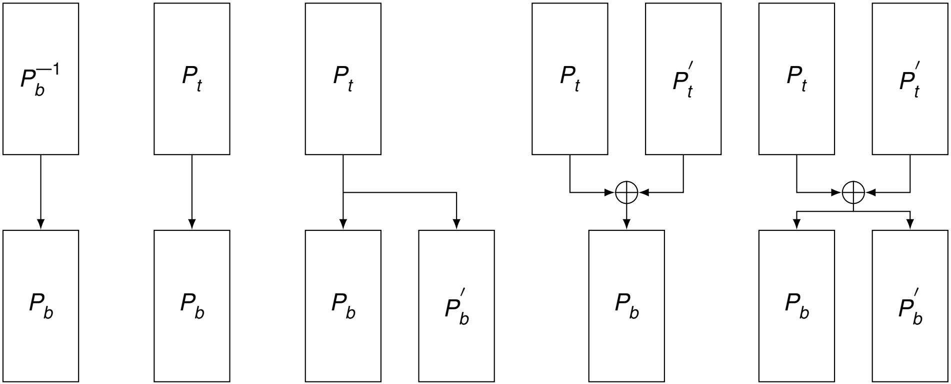
$$\tilde{E}(K, i, N, (\tilde{E})^{-1}(K, 0, N, X))$$

where $i \in \mathscr{I} \setminus \{0\}$*. Then, ...*

$$\mathsf{Adv}_{\Gamma}^{\mathsf{tprp}}(\mathbf{G}) \leq \mathsf{Adv}_{\tilde{E}}^{\mathsf{stprp}}(\mathbf{G}^{'})$$

3. Give the adversary oracle access to the stream cipher and reduce the security to that of the NaT MAC.

Cryptanalysis of LLSIV



Skinny Cryptanalysis

Model	Technique	Ref.	Number of Rounds	Data	Time
Single Key	ID	[TAY17]	22	$2^{92.22}$	$2^{373.48}$
	MitM	[DHS+21]	23	2^{120}	2^{368}
	DS-MitM	[SSS+23]	23	2^{96}	2^{372}
	Diff-MitM	[BDD+23]	25	$2^{122.3}$	$2^{372.5}$
Chosen Tweak	Int	[HSE23]	26	2^{121}	2^{344}
	DS-MitM	[SSS+23]	25	2^{96}	$2^{363.83}$
Related Key	Rectangle	[HBS21]	30	2^{125}	2^{361}
		[QDW+21]	30	2^{122}	2^{341}
		[DQSW22]	32	2^{123}	2^{355}
		[SZY+22]	32	2^{123}	2^{345}

LUND UNIVERSITY

Claims using Skinny: pLLSIV

Scheme	Max. Length	Data	Time	Key Size
pLLSIV ($r = 25$)	2^{16}	2^{46}	2^{112}	2×128
LLSIV ($r = 40$)	2^{64}	$2^{128}/\mu$	2^{128}	$192 + 128$

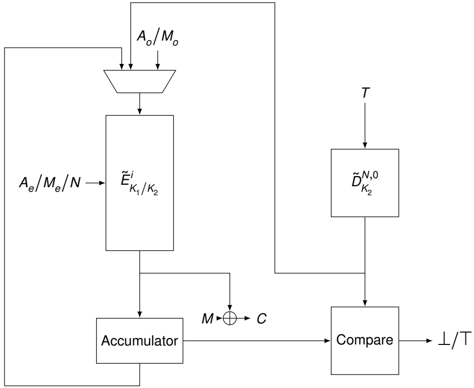
- It is possible to reduce the key size with domain separation: out of scope.

Comparison

LUND UNIVERSITY

Scheme	Model	TBC Encryption	TBC Decryption	# cycles
SCT-2k	Single-Core	Iterative	-	$r(m+1)$
LLSIV	Single-Core	Iterative	Iterative	$r(m+1)$
pLLSIV	Single-Core	Iterative	Iterative	$pr(m+1)$
SCT-2k	Multi-Core	Multi-core	-	$r(\lceil m/c \rceil + 1)$
LLSIV	Multi-Core	Multi-core	Iterative	$r\lceil (m+1)/c \rceil$
pLLSIV	Multi-Core	Multi-core	Iterative	$pr\lceil (m+1)/c \rceil$
SCT-2k	Pipelined	Pipelined	-	$2r+m-1$
LLSIV	Pipelined	Pipelined	Iterative	$r+m$
pLLSIV	Pipelined	Pipelined	Iterative	$pr+m$

Hardware Implementation



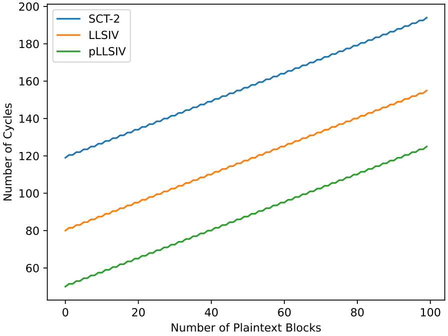
Pipelined Implementation

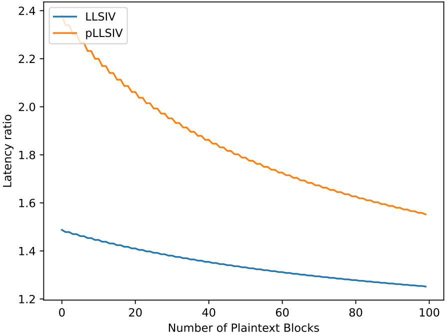
LUND UNIVERSITY

Table: Synthesis results of the pipelined implementations of SCT-2k, LLSIV and pLLSIV on the Xilinx Kintex-7 FPGA. a and m are the number of 128-bit blocks of associated data and plaintext, respectively. The number of cycles is for the encryption algorithm.

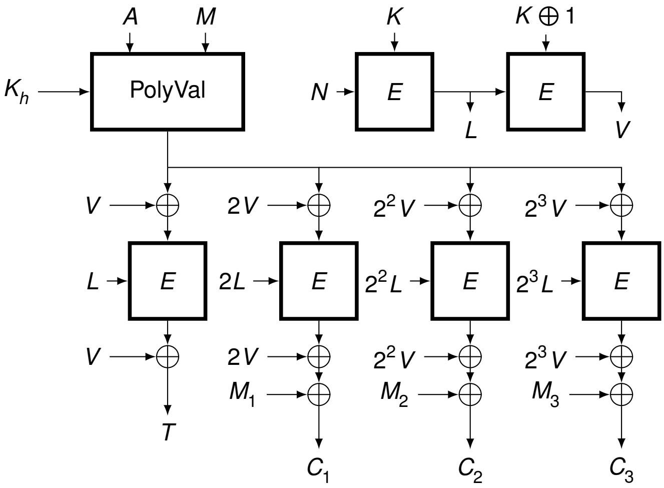
Scheme	LUTs	Flip Flops	# of Cycles
SCT-2k	8230	20581	$118 + a/2 + 3m/2$
LLSIV	9243	20587	$79 + a/2 + 3m/2$
pLLSIV	5392	12907	$49 + a/2 + 3m/2$

Latency


SCT-2
LLSIV
pLLSIV
Number of Cycles
200
180
160
140
120
100
80
60
0
20
40
60
80
100
Number of Plaintext Blocks


LLSIV
pLLSIV
Latency ratio
2.4
2.2
2.0
1.8
1.6
1.4
1.2
0
20
40
60
80
100
Number of Plaintext Blocks

AES-Based Instance: AES-POLYVAL-ICE2



Pipelined Implementation

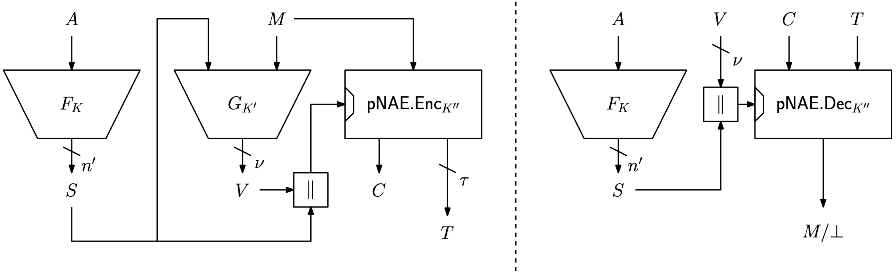
LUND UNIVERSITY

Table: Synthesis results of the pipelined implementations of AES-GCM-SIV and LLSIV-PolyVal-ICE2 on the Xilinx Kintex-7 FPGA. a and m are the number of 128-bit blocks of associated data and plaintext, respectively. The number of cycles is for the encryption algorithm.

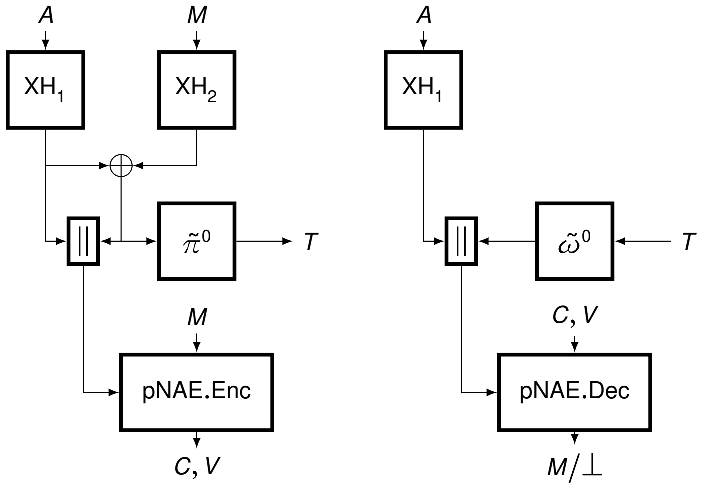
Scheme	LUTs	Flip Flops	# of Cycles
AES-GCM-SIV	12780	3017	$4(a+m)+35+m$
LLSIV-PolyVal-ICE2	13965	3401	$4(a+m)+10+m$

Decryption Fast SIV

- Proposed by Minematsu in 2020.



LLDFV



Security

1. No collision on the concatenated hashes in encryption queries.
2. Nonces in decryption queries never appear during encryption queries → need strong TPRP security.
3. Then, the security falls back on the security of pNAE (nonce-based AE without A).

LUND UNIVERSITY

Questions?

Thanks for listening.

More details at `https://eprint.iacr.org/2024/550`

Christina Boura, Nicolas David, Patrick Derbez, Gregor Leander, and María Naya-Plasencia.
Differential meet-in-the-middle cryptanalysis.
In *Annual International Cryptology Conference*, pages 240–272. Springer, 2023.

Xiaoyang Dong, Jialiang Hua, Siwei Sun, Zheng Li, Xiaoyun Wang, and Lei Hu.
Meet-in-the-middle attacks revisited: key-recovery, collision, and preimage attacks.
In *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part III 41*, pages 278–308. Springer, 2021.

Xiaoyang Dong, Lingyue Qin, Siwei Sun, and Xiaoyun Wang.
Key guessing strategies for linear key-schedule algorithms in rectangle attacks.
Springer-Verlag, 2022.

Hosein Hadipour, Nasour Bagheri, and Ling Song.
Improved rectangle attacks on skinny and craft.

IACR Transactions on Symmetric Cryptology, pages 140–198, 2021.

Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder.
Finding the impossible: Automated search for full impossible-differential, zero-correlation, and integral attacks.
In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 128–157. Springer, 2023.

Lingyue Qin, Xiaoyang Dong, Xiaoyun Wang, Keting Jia, and Yunwen Liu.
Automated search oriented to key recovery on ciphers with linear key schedule: applications to boomerangs in skinny and forkskinny.
IACR Transactions on Symmetric Cryptology, pages 249–291, 2021.

Danping Shi, Siwei Sun, Ling Song, Lei Hu, and Qianqian Yang.
Exploiting non-full key additions: full-fledged automatic demirci-selcuk meet-in-the-middle cryptanalysis of skinny.

In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 67–97. Springer, 2023.

Ling Song, Nana Zhang, Qianqian Yang, Danping Shi, Jiahao Zhao, Lei Hu, and Jian Weng.
Optimizing rectangle attacks: a unified and generic framework for key recovery.
In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 410–440. Springer, 2022.

Mohamed Tolba, Ahmed Abdelkhalek, and Amr M Youssef.
Impossible differential cryptanalysis of reduced-round skinny.
In *International Conference on Cryptology in Africa*, pages 117–134. Springer, 2017.