LOTUS and LOCUS AEAD: Hardware Benchmarking and Security Analysis

Avik Chakraborti[1], Nilanjan Datta[2], Ashwin Jha[2], Cuauhtemoc Mancillas Lopez[3], Mridul Nandi[2], Yu Sasaki[1]

[1] NTT Secure Platform Laboratories, Japan
[2] Indian Statistical Institute, Kolkata, India
[3] Computer Science Department, CINVESTAV-IPN, Mexico

chakraborti.avik@lab.ntt.co.jp,nilanjan_isi_jrf@yahoo.com,ashwin.jha1991@gmail.com,cuauhtemoc.mancillas83@gmail.com,mridul.nandi@gmail.com,sasaki.yu@lab.ntt.co.jp

Abstract. In this short note, we propose two new designs for lightweight AE modes, called LOCUS and LOTUS, structurally similar to OCB and OTR, respectively. These modes achieve notably higher AE security bounds with lighter primitives (only a 64-bit tweakable block cipher). Especially, they satisfy the NIST requirements: achieving security against an adversary that can make close to 2^{64} queries and 2^{128} computations, even when instantiated with a 64-bit primitive with 128-bit key. Both these modes are fully parallelizable and provide full INT-RUP security. We use TweGIFT-64, a tweakable variant of the GIFT block cipher, to instantiate our AE modes. TweGIFT-64-LOCUS and TweGIFT-64-LOTUS are significantly light in hardware implementation. To justify, we provide our FPGA based implementation results, which demonstrate that TweGIFT-64-LOCUS consumes only 257 slices and 690 LUTs, while TweGIFT-64-LOTUS consumes only 255 slices and 664 LUTs. We have also provided concrete security analysis both OCB and OTR.

Keywords: OCB · OTR · TweGIFT · lightweight · INT-RUP · ideal cipher

1 Specification

In this section, we present the specifications of LOTUS and LOCUS authenticated encryption mode [10]. The encryption algorithm of both LOTUS and LOCUS modes receives an encryption key $K \in \{0,1\}^{\kappa}$, a nonce $N \in \{0,1\}^{\kappa}$, an associated data $A \in \{0,1\}^{*}$, and a message $M \in \{0,1\}^{*\leftharpoonup}$as inputs, and returns a ciphertext $C \in \{0,1\}^{|M|}$, and a tag $T \in \{0,1\}^{n}$. The complete specification of LOTUS and LOCUS authenticated encryption is given in Algorithm 1 and 2 respectively.

2 Hardware Implementation

In this section, we provide a brief idea on the FPGA implementations of our designs. We first briefly describe our hardware implementation details of the TweGIFT-64 module. We have implemented TweGIFT-64 on Virtex 6 (target device xc6vlx760) using the RTL approach and a basic iterative type architecture. We would like to emphasize that our implementation is round based and it uses 64-bit data path, a smaller implementation can be obtained using smaller datapaths 4-bit, 8-bit, 16-bit or even serialized implementations.

Table 1 provides the implementation details of TweGIFT-64 on Virtex 6. It is evident from the results that the difference in the number of LUTs is 119 (caused by the inclusion of the decryption rounds and the multiplexers to select the input to the state register). The difference in terms of the number of slices is about 36 such that one slice in Virtex 6 has 4 LUTs and 2 Flip-flops (depends how a design is optimized and placed by the Xilinx tools).

Table 1: TweGIFT-64 Implemented FPGA Results on Virtex 6

Mode	# Slice Registers	# LUTs	# Slices	Frequency (MHZ)	Gbps	Mbps/ LUT	Mbps/ Slice
Enc/dec	273	734	270	425.99	0.94	1.28	3.48
Enc	275	333	134	540.56	1.19	3.57	8.88

Table 2: TweGIFT-64 Implemented FPGA Results on Virtex 7

Platform	# Slice Registers	# LUTs	# Slices	Frequency (MHZ)	Gbps	Mbps/ LUT	Mbps/ Slice
Enc/dec	273	730	265	441.71	0.97	1.32	3.66
Enc	275	329	134	554.32	1.22	3.71	9.10

Algorithm 1 The encryption algorithm of LOTUS.

```
function LOTUS.enc$(K, N, A, M)$
    $C \quad \perp, W_\oplus \quad 0, V_\oplus \quad 0$
    $(K_N, \Delta_N) \quad$ init$(K, N)$
    if $|A| \neq 0$ then
        $(K_N, V_\oplus) \quad$ proc_ad$(K_N, \Delta_N, A)$
    if $|M| \neq 0$ then
        $(K_N, W_\oplus, C) \quad$ proc_pt$(K_N, \Delta_N, M)$
    $T \quad$ proc_tg$(K_N, \Delta_N, V_\oplus, W_\oplus)$
    return $(C, T)$

function init$(K, N)$
    $Y \quad \widetilde{\mathsf{E}}^0_K(0^n)$
    $K_N \quad K \oplus \leftarrow N$
    $\Delta_N \quad \widetilde{\mathsf{E}}^1_{K_N}(Y)$
    return $(K_N, \Delta_N)$

function proc_ad$(K_N, \Delta_N, A)$
    $L \quad K_N$
    $(A_a, \dots, A_1) \quad ^{n} \quad A$
    for $i = 1$ to $a - 1$ do
        $X \quad A_i \oplus \Delta_N$
        $L \quad L \odot \alpha$
        $V \quad \widetilde{\mathsf{E}}^2_L(X)$
        $V_\oplus \quad V_\oplus \oplus \leftarrow V$
    $X \quad$ ozs$(A_a) \oplus \Delta_N$
    $L \quad L \odot \alpha$
    if $|A_a| = n$ then
        $V \quad \widetilde{\mathsf{E}}^2_L(X)$
    else
        $V \quad \widetilde{\mathsf{E}}^3_L(X)$
    $V_\oplus \quad V_\oplus \oplus V$
    return $(L, V_\oplus)$

function proc_pt$(K_N, \Delta_N, M)$
    $L \quad K_N$
    $(M_m, \dots, M_1) \quad ^{n} \quad M$
    $d = \lceil m/2 \rceil \leftarrow$
    for $i = 1$ to $d - 1$ do
        $j = 2i - 1$
        $X_1 \quad M_j \oplus \Delta_N$
        $L \quad L \odot \leftarrow \alpha$
        $W_1 \quad \widetilde{\mathsf{E}}^4_L(X_1)$
        $Y_1 \quad \widetilde{\mathsf{E}}^4_L(W_1)$
        $X_2 \quad Y_1 \oplus \leftarrow M_{j+1}$
        $W_2 \quad \widetilde{\mathsf{E}}^5_L(X_2)$
        $Y_2 \quad \widetilde{\mathsf{E}}^5_L(W_2)$
        $W_\oplus \quad W_\oplus \oplus W_1 \oplus W_2$
        $C_j \quad X_2 \oplus \Delta_N$
        $C_{j+1} \quad X_1 \oplus Y_2$
    $X_1 \quad \langle |M| - 2(d-1)n \rangle_n \oplus \Delta_N$
    $L \quad L \odot \alpha$
    $W_1 \quad \widetilde{\mathsf{E}}^{12}_L(X_1)$
    $Y_1 \quad \widetilde{\mathsf{E}}^{12}_L(W_1)$
    $X_2 \quad Y_1 \oplus M_{2d-1}$
    $C_{2d-1} \quad$ chop$(X_2 \oplus \Delta_N, |M_{2d-1}|)$
    $W_\oplus \quad W_\oplus \oplus W_1$
    $C \quad (C_{2d-1}, \dots, C_1)$
    if $2d = m$ then
        $W_2 \quad \widetilde{\mathsf{E}}^{13}_L(X_2)$
        $W_\oplus \quad W_\oplus \oplus W_2$
        $Y_2 \quad \widetilde{\mathsf{E}}^{13}_L(W_2)$
        $C_{2d} \quad \lfloor X_1 \oplus Y_2 \rfloor_{|M_{2d}|} \oplus M_{2d}$
        $C \quad C_{2d} \| C$
    $W_\oplus \quad W_\oplus \oplus M_m$
    return $(L, W_\oplus, C)$

function proc_tg$(K_N, \Delta_N, V_\oplus, W_\oplus)$
    $L \quad K_N \odot \alpha$
    if $(\lceil |A|/n \rceil \leftarrow + \lceil |M|/n \rceil) \mod 2 = 0$ then
        $X_\oplus \quad V_\oplus \oplus W_\oplus \oplus \Delta_N$
    else
        $X_\oplus \quad V_\oplus \oplus W_\oplus$
    $T \quad \widetilde{\mathsf{E}}^6_L(X_\oplus) \oplus \Delta_N$
    return $T$
```

Algorithm 2 The encryption algorithm of LOCUS. The subroutine proc_ad and proc_tag are identical to the one used in LOTUS.

```
function LOCUS.enc$(K, N, A, M)$
    $C \quad \perp, \; W_\oplus \quad 0, \; V_\oplus \quad 0$
    $(K_N, \Delta_N) \quad$ init$(K, N)$
    if $|A| \neq 0$ then
        $(K_N, V_\oplus) \quad$ proc_ad$(K_N, \Delta_N, A)$
    if $|M| \neq 0$ then
        $(K_N, W_\oplus, C) \quad$ proc_pt$(K_N, \Delta_N, M)$
    $T \quad$ proc_tg$(K_N, \Delta_N, V_\oplus, W_\oplus)$
    return $(C, T)$

function proc_pt$(K_N, \Delta_N, M)$
    $L \quad K_N$
    $(M_m, \dots, M_1) \quad ^{n} \quad M$
    for $j = 1$ to $m - 1$ do
        $X \quad M_j \oplus \Delta_N$
        $L \quad L \odot \leftarrow \alpha$
        $W \quad \widetilde{\mathsf{E}}^4_L(X)$
        $W_\oplus \quad W_\oplus \oplus W$
        $Y \quad \widetilde{\mathsf{E}}^4_L(W)$
        $C_j \quad Y \oplus \Delta_N$
    $L \quad L \odot \alpha$
    $X \quad \langle |M_m| \rangle_n \oplus \Delta_N$
    $W \quad \widetilde{\mathsf{E}}^5_L(X)$
    $Y \quad \widetilde{\mathsf{E}}^5_L(W)$
    $C_m \quad \lfloor Y \oplus \Delta_N \rfloor_{|M_m|} \oplus \leftarrow M_m$
    $W_\oplus \quad W_\oplus \oplus W \oplus \leftarrow M_m$
    $C \quad (C_m, \dots, C_1)$
    return $(L, W_\oplus, C)$
```

2.1 Implementation of LOCUS and LOTUS

The hardware implementations of LOCUS and LOTUS are written in VHDL and are implemented on both Virtex 6 xc6vlx760 and Virtex 7 xc7vx415t. We use the RTL approach and use a basic round based architecture. The areas are provided in terms of the number of slice registers, slice LUTs and the number of occupied slices. The detailed implementation results are depicted in Table 3.

Table 3: LOCUS and LOTUS (combined enc/dec circuit) Implemented FPGA Results.

Platform	**Scheme**	**# Slice Registers**	**# LUTs**	**# Slices**	**Frequency (MHZ)**	**Throughput (Gbps)**	**Mbps/ LUT**	**Mbps/ Slice**
Virtex 6	LOCUS	437	1146	418	348.67	0.39	0.34	0.94
Virtex 7	LOCUS	430	1154	439	392.20	0.44	0.38	1.00
Virtex 6	LOCUS-e	427	698	250	368.34	0.41	0.59	1.65
Virtex 7	LOCUS-e	424	704	272	406.84	0.46	0.65	1.68
Virtex 6	LOTUS	571	868	317	351.25	0.39	0.45	1.24
Virtex 7	LOTUS	565	865	317	424.45	0.48	0.55	1.50
Virtex 6	LOTUS-e	564	801	251	380.84	0.43	0.53	1.70
Virtex 7	LOTUS-e	564	800	249	414.42	0.47	0.58	1.87
Virtex 6	LOTUS-d	566	804	245	379.83	0.43	0.53	1.74
Virtex 7	LOTUS-d	563	791	254	418.91	0.47	0.59	1.85

2.2 Benchmarking LOCUS and LOTUS

In this section, we provide a benchmark of hardware implementation results for both LOCUS and LOTUS with the ATHENa listed results in [4, 3] on both Virtex 6 and 7. We would like to point out that our implementations ignore the API area overheads (as mentioned in [20, 22]) related to the CAESAR API (which is update of the GMU hardware API). Nevertheless, the result shows that both our implementations consume a very low hardware footprint and achieve highly competitive results, even if we add the overhead associated to the CAESAR API. A detailed comparison can be found below in Table 4 and 5. Note that, the hardware areas for SUNDAE [6] is given in GEs (ASIC platform). Hence, we do not include these results in the table. The comparison table shows that our implementation results are highly competitive and one of the best in the literature.

3 Security Analysis of LOCUS and LOTUS

Before delving into the security proofs, we give an alternative formulation for LOCUS and LOTUS based on a tweakable block cipher. This formulation extends Rogaway's XEX [25] based abstraction of OCB.

3.1 Θ-LOCUS and Θ-LOTUS

Let $\mathcal{T} \Leftarrow \{0,1\}^\kappa \times \{2,3,\ldots,15\} \times [2^n]$ and $\widetilde{\Pi} \quad {}_{\$}\mathrm{TPerms}(\mathcal{T},\{0,1\}^n)$. We define two new authenticated encryption schemes Θ-LOTUS[$\widetilde{\Pi}$] and Θ-LOCUS[$\widetilde{\Pi}$] in Algorithms 3 and 4, respectively.

Notice that the modified algorithms are implicitly keyed due to the tweakable random permutation $\widetilde{\Pi}$.

Let $\widetilde{\mathsf{E}}$ be a tweakable ideal cipher over key space $\{0,1\}^\kappa$, tweak space (15], and block space $\{0,1\}^n$. Now, we define $\widetilde{\mathsf{P}}$ as a tweakable block cipher over key space $\{0,1\}^\kappa$, tweak space $\mathcal{T}$, and block space $\{0,1\}^n$, by the following mapping: $\forall(K,N,d,i,X) \in \mathcal{T} \leftrightarrow\!\times \{0,1\}^n$,

$$\widetilde{\mathsf{P}}_K^{N,d,i}(X) := \widetilde{\mathsf{E}}_{L_i}^{d}(X \oplus \Delta_N) \oplus \Delta_N. \quad (1)$$

where $L_i = 2^i(K \oplus N)$ and $\Delta_N = \widetilde{\mathsf{E}}^1_{K\oplus N}(\widetilde{\mathsf{E}}^0_K(0))$.

This definition, though artificial in nature, serves its purpose well. Notably, we can now view LOTUS and LOCUS as instantiations of Θ-LOTUS and Θ-LOCUS, namely, Θ-LOTUS[$\widetilde{\mathsf{P}}[\widetilde{\mathsf{E}}]$] and Θ-LOCUS[$\widetilde{\mathsf{P}}[\widetilde{\mathsf{E}}]$], respectively. Later on we argue the security of LOCUS and LOTUS under this modified view.

We remark here that the small modifications in the specification of LOTUS and LOCUS (see section 1) are introduced precisely to exploit this modularity. As we see later in this section, these changes make the proof modular and much easier to understand. The security of the original construction as given in the NIST submission [11] is exactly the same, though requires a more dedicated and notationally complex proof.

3.2 Combined Security Notion for Integrity under RUP and Privacy

Let Φ be a nonce-based authenticated encryption scheme. Conventionally, the AE security, which suffices for privacy as well as integrity, of Φ is argued through indistinguishability game where an adversary tries to distinguish the real oracle $\mathcal{R} \Leftarrow (\Phi.\mathsf{enc}, \Phi.\mathsf{ver})$ from the ideal oracle $\mathcal{I} \Leftarrow (\$_{\mathsf{e}}, \bot)$.

We extend this method to argue integrity under RUP and privacy of Φ via an indistinguishability game where the adversary tries to distinguish the real oracle $\mathcal{R} \Leftarrow (\Phi.\mathsf{enc}, \Phi.\mathsf{dec}, \Phi.\mathsf{ver})$ and the ideal oracle $\mathcal{I} \Leftarrow (\$_{\mathsf{e}}, \$_{\mathsf{d}}, \bot)$. In case

Algorithm 3 The encryption algorithm of Θ-LOTUS$[\widetilde{\Pi}]$.

```
function Θ-LOTUS[Π̃].enc($N, A, M$)
    $C \quad \perp, W_\oplus \quad 0, V_\oplus \quad 0$
    $\lceil |A|/n \rceil \Leftarrow a$
    $\lceil |M|/n \rceil \Leftarrow m$
    if $a \neq 0$ then
        $V_\oplus \quad$ proc_ad($A$)
    if $m \neq 0$ then
        $(W_\oplus, C) \quad$ proc_pt($M$)
    $T \quad$ proc_tg($V_\oplus, W_\oplus$)
    return $(C, T)$

function proc_ad($A$)
    $(A_a, \ldots, A_1) \overset{n}{\quad} A$
    for $i = 1$ to $a - 1$ do
        $V_i \quad \widetilde{\Pi}^{(N,2,i)}(A_i)$
        $V_\oplus \quad V_\oplus \oplus V_i$
    if $|A_a = n|$ then
        $V_a \quad \widetilde{\Pi}^{(N,2,a)}(\mathsf{ozs}(A_a))$
    else
        $V_a \quad \widetilde{\Pi}^{(N,3,a)}(\mathsf{ozs}(A_a))$
    $V_\oplus \quad V_\oplus \oplus V_a$
    return $V_\oplus$
```

```
function proc_pt($M$)
    $(M_m, \ldots, M_1) \overset{n}{\quad} M$
    $d = \lceil m/2 \rceil \leftarrow$
    for $i = 1$ to $d - 1$ do
        $j = 2i - 1$
        $W_j \quad \widetilde{\Pi}^{(N,4,a+i)}(M_j)$
        $C_j \quad \widetilde{\Pi}^{(N,4,a+i)}(W_j) \oplus M_{j+1}$
        $W_{j+1} \quad \widetilde{\Pi}^{(N,5,a+i)}(C_j)$
        $C_{j+1} \quad \widetilde{\Pi}^{(N,5,a+i)}(W_{j+1}) \oplus M_j$
        $W_\oplus \quad W_\oplus \oplus W_1 \oplus W_2$
    $X \quad \langle |M| - 2(d-1)n \rangle_n$
    $W_{2d-1} \quad \widetilde{\Pi}^{N,12,a+d}(X)$
    $Y \quad \widetilde{\Pi}^{N,12,a+d}(X) \oplus \mathsf{ozs}(M_{2d-1})$
    $C_{2d-1} \quad \lfloor Y \rfloor_{|M_{2d-1}|}$
    $W_\oplus \quad W_\oplus \oplus W_{2d-1}$
    $C \quad (C_{2d-1}, \ldots, C_1)$
    if $2d = m$ then
        $W_{2d} \quad \widetilde{\Pi}^{(N,13,d)}(Y)$
        $W_\oplus \quad W_\oplus \oplus W_{2d}$
        $C_{2d} \quad \lfloor \widetilde{\Pi}^{(N,13,d)}(W_{2d}) \oplus X \rfloor |M_{2d}|) \oplus M_{2d}$
        $C \quad C_{2d} \| C$
    $W_\oplus \quad W_\oplus \oplus M_{m-1}$
    return $(W_\oplus, C)$

function proc_tg($V_\oplus, W_\oplus$)
    $X_\oplus \quad V_\oplus \oplus W_\oplus$
    $T \quad \widetilde{\Pi}^{N,6,a+m}(X_\oplus)$
    return $T$
```

Algorithm 4 The encryption algorithm of Θ-LOCUS$[\widetilde{\Pi}]$. The subroutine proc_ad and proc_tg are identical to the one used in Θ-LOTUS$[\widetilde{\Pi}]$.

```
function Θ-LOCUS[Π̃].enc($N, A, M$)
    $C \quad \perp, \ W_\oplus \quad 0, \ V_\oplus \quad 0$
    if $|A| \neq 0$ then
        $V_\oplus \quad$ proc_ad($A$)
    if $|M| \neq 0$ then
        $(W_\oplus, C) \quad$ proc_pt($M$)
    $T \quad$ proc_tg($V_\oplus, W_\oplus, |A| + |M|$)
    return $(C, T)$
```

```
function proc_pt($M$)
    $(M_m, \ldots, M_1) \overset{n}{\quad} M$
    for $j = 1$ to $m - 1$ do
        $W_j \quad \widetilde{\Pi}^{(N,4,j)}(M_j)$
        $W_\oplus \quad W_\oplus \oplus W_j$
        $C_j \quad \widetilde{\Pi}^{(N,4,j)}(W_j)$
    $X \quad \langle |M_m| \rangle_n$
    $W_m \quad \widetilde{\Pi}^{(N,5,j)}(X)$
    $W_\oplus \quad W_\oplus \oplus W_m \oplus M_m$
    $Y \quad \widetilde{\Pi}^{(N,5,j)}(W_m)$
    $C_m \quad \lceil Y \rceil_{|} M_m| \oplus M_m$
    $C \quad (C_m, \ldots, C_1)$
    return $(W_\oplus, C)$
```

Table 4: Comparison on Virtex 6 [4]. Here BC denotes block cipher, SC denotes Streamcipher, (T)BC denotes (Tweakable) block cipher and BC-RF denotes the block cipher's round function,'-' means that the data is not available.

Scheme	Underlying Primitive	# LUTs	# Slices	Gbps	Mbps/ LUT	Mbps/ Slice
LOCUS	BC (non AES)	1146	418	0.39	0.34	0.94
LOTUS	BC (non AES)	868	317	0.39	0.45	1.24
AES-OTR [23]	BC	5102	1385	2.741	0.537	1.979
AES-OCB [21]	BC	4249	1348	3.122	0.735	2.316
AES-OCB [21]	BC	4249	1348	1.56	0.37	1.16
AES-GCM [17]	BC	3175	1053	3.239	1.020	3.076
AES-COPA [2]	BC	7754	2358	2.500	0.322	1.060
CLOC-AES [18]	BC	3145	891	2.996	0.488	1.724
CLOC-TWINE [18]	BC (non-AES)	1689	532	0.343	0.203	0.645
ELmD [15]	BC	4302	1584	3.168	0.736	2.091
JAMBU-AES [27]	BC	1836	652	1.999	1.089	3.067
JAMBU-SIMON [27]	BC (non-AES)	1222	453	0.363	0.297	0.801
SILC-AES [18]	BC	3066	921	4.040	1.318	4.387
SILC-LED [18]	BC (non-AES)	1685	579	0.245	0.145	0.422
SILC-PRESENT [18]	BC (non-AES)	1514	548	0.407	0.269	0.743
COFB-AES [13, 13]	BC	1075	442	2.850	2.240	6.450
AEGIS [28]	BC-RF	7592	2028	70.927	9.342	34.974
DEOXYS [19]	TBC	3143	951	2.793	0.889	2.937
Beetle[Light+] [12]	Sponge	616	252	1.879	3.050	7.369
Beetle[Secure+] [12]	Sponge	998	434	2.520	2.525	5.806
ASCON-128 [16]	Sponge	1271	413	3.172	2.496	7.680
Ketje-Jr [7]	Sponge	1236	412	2.832	2.292	6.875
NORX [5]	Sponge	2964	1016	11.029	3.721	10.855
PRIMATES-HANUMAN [1]	Sponge	1012	390	0.964	0.953	2.472
ACORN [26]	SC	455	135	3.112	6.840	23.052
TriviA-ck [8, 9, 14]	SC	2118	687	15.374	7.259	22.378

of $\mathcal{I}$, $\$_{\mathsf{d}}$ is an arbitrary interface for decryption in the ideal world that should mimic the decryption algorithm of the authenticated encryption scheme at hand. Now, we define the NAEAD with integrity in RUP, we call it NAEAD*, advantage of some adversary $\mathscr{A}$ against an AEAD scheme Φ as

$$\mathbf{Adv}_{\Phi}^{\mathsf{naead}^\star}(\mathscr{A}) := \left|\Pr[\mathscr{A}^{\mathcal{R}} = 1] - \Pr[\mathscr{A}^{\mathcal{I}} = 1]\right|, \tag{2}$$

$$\mathbf{Adv}_{\Phi}^{\mathsf{naead}^\star} := \max_{\mathscr{A}} \mathbf{Adv}_{\Phi}^{\mathsf{naead}^\star}(\mathscr{A}). \tag{3}$$

One can easily argue that this modified indistinguishability game implies both integrity under RUP and privacy.

First, if there exist an adversary $\mathscr{D}$ that can break the privacy security of Φ, then one can construct an adversary $\mathscr{A}$ that can distinguish $\mathcal{R}$ from $\mathcal{I}$ with at least the privacy advantage of $\mathscr{D}$. This can be easily argued as $\mathscr{A}$ has access to one of $\Phi.\mathsf{enc}$ or $\$_{\mathsf{e}}$, whence it can rightly simulate the oracle access for $\mathscr{D}$. Finally, $\mathscr{A}$ returns 1 if $\mathscr{D}$ returns 1, and $\mathscr{A}$ returns a bit chosen uniform at random if $\mathscr{D}$ returns 0. Clearly, we have

$$\mathbf{Adv}_{\Phi}^{\mathsf{priv}}(\mathscr{D}) \leq \mathbf{Adv}_{\Phi}^{\mathsf{naead}^\star}(\mathscr{A}).$$

Second, if the oracle $\mathcal{R}$ is distinguishable from $\mathcal{I}$ with at most $\epsilon = \mathbf{Adv}_{\Phi}^{\mathsf{naead}^\star}$ then the oracle $\mathcal{R}' = (\Phi.\mathsf{enc}, \Phi.\mathsf{dec}, \perp)$ is also distinguishable from $\mathcal{I}$ with at most ϵ, as we are actually removing adversary's access to $\Phi.\mathsf{ver}$. Then, for some adversary $\mathscr{F}$, we have

$$\begin{aligned}\mathbf{Adv}_{\Phi}^{\mathsf{int\text{-}rup}}(\mathscr{F}) &= \left|\Pr[\mathscr{F}^{\mathcal{R}} = 1] - \Pr[\mathscr{F}^{\mathcal{R}'} = 1]\right| \\ &\leq \left|\Pr[\mathscr{F}^{\mathcal{R}} = 1] - \Pr[\mathscr{F}^{\mathcal{I}} = 1]\right| + \left|\Pr[\mathscr{F}^{\mathcal{I}} = 1] - \Pr[\mathscr{F}^{\mathcal{R}'} = 1]\right| \\ &\leq 2\mathbf{Adv}_{\Phi}^{\mathsf{naead}^\star}.\end{aligned}$$

3.3 NAEAD* Security of LOCUS

First, we give the definition of the ideal oracle decryption interface $\$_{\mathsf{d}}$. Our main goal is to keep the interfaces $(\$_{\mathsf{e}}, \$_{\mathsf{d}})$ as close to $(\Theta\text{-}\mathsf{LOCUS.enc}, \Theta\text{-}\mathsf{LOCUS.dec})$ as possible. To motivate the rationale behind our choice of $\$_{\mathsf{d}}$, we first redefine $\$_{\mathsf{e}}$ as follows:

Table 5: Comparison on Virtex 7 [4].

Scheme	# LUTs	# Slices	Gbps	Mbps/ LUT	Mbps/ Slice
LOCUS	1154	439	0.44	0.38	1.00
LOTUS	865	317	0.48	0.55	1.50
AES-OTR	4263	1204	3.187	0.748	2.647
OCB	4269	1228	3.608	0.845	2.889
AES-COPA	7795	2221	2.770	0.355	1.247
AES-GCM	3478	949	3.837	1.103	4.043
CLOC-AES	3552	1087	3.252	0.478	1.561
CLOC-TWINE	1552	439	0.432	0.278	0.984
SILC-AES	3040	910	4.365	1.436	4.796
SILC-LED	1682	524	0.267	0.159	0.510
SILC-PRESENT	1514	484	0.479	0.316	0.990
ELmD	4490	1306	4.025	0.896	3.082
JAMBU-AES	1595	457	1.824	1.144	3.991
JAMBU-SIMON	1200	419	0.368	0.307	0.878
COFB-AES	1456	555	2.820	2.220	5.080
SAEB [24]	348	–	–	–	–
AEGIS	7504	1983	94.208	12.554	47.508
DEOXYS	3234	954	1.472	0.455	2.981
Beetle[Light+]	608	312	2.095	3.445	6.715
Beetle[Secure+]	1101	512	2.993	2.718	5.846
ASCON-128	1373	401	3.852	2.806	9.606
Ketje-Jr	1567	518	4.080	2.604	7.876
NORX	2881	857	10.328	3.585	12.051
PRIMATES-HANUMAN	1148	370	1.072	0.934	2.897
ACORN	499	155	3.437	6.888	22.174
TriviA-ck	2221	684	14.852	6.687	21.713

```
1. Initialize $\widetilde{\Pi} \quad {}_{\$}\,\mathrm{TPerms}(\mathcal{N} \times \mathbb{N} \times \{0,1\}, \{0,1\}^n)$ and $\Gamma \quad {}_{\$}\,\mathrm{Funcs}(\mathcal{N} \times \mathcal{A} \times \mathcal{M}, \{0,1\}^n)$.
2. On input $(N, A, M)$ do:
        $a \quad \lceil |A|/n \rceil,\ \ell \quad \lceil |M|/n \rceil$.
        $(M_1, \ldots, M_\ell) \quad ^{n} \quad M$.
        $\forall i \in [\ell - 1],\ C_i \quad \widetilde{\Pi}^{(N,a+i,0)}(M_i)$.
        $C_\ell \quad \lfloor \widetilde{\Pi}^{(N,a+\ell,1)}(\langle |M_\ell| \rangle_n) \rfloor_{|M_\ell|} \oplus M_\ell$.
        $C \quad C_1 \| \cdots \| C_\ell$.
        $T \quad \Gamma(N, A, M)$.
        Return $(C, T)$.
```

Observe that for nonce-respecting adversary the modified definition of $\$_{\mathsf{e}}$ is identical to the usual random string definition. This is because each call to the underlying tweakable random permutation is made with a different tweak, whence the ciphertext blocks are uniform at random and independent of each other. Further, Γ is independent of $\widetilde{\Pi}$, whence tag is uniform at random and independent of the ciphertext blocks.

We remark that the ciphertext generation is similar to the classical ECB mode, where each block is encrypted by $\widetilde{\Pi}$ with the nonce and block index playing the role of tweak value. With this tweakable ECB mode in mind, we can easily define $\$_{\mathsf{d}}$ as the inverse of the tweakable ECB mode. Further, note that the $\$_{\mathsf{e}}$ and $\$_{\mathsf{d}}$ definitions are quite similar to the Θ-LOCUS[$\widetilde{\Pi}$] definition. Here we use one $\widetilde{\Pi}$ call instead of two to generate the ciphertext block.

The main technical result on the security of LOCUS is given in Theorem 1.

Theorem 1. *For any nonce-respecting $(q_e, q_d, q_v, q_p, \sigma_e, \sigma_d, \sigma_v)$-adversary $\mathscr{A}$, we have*

$$\mathbf{Adv}^{\mathsf{naead}^\star}_{\mathsf{LOCUS}[\widetilde{\mathsf{E}}]}(\mathscr{A}) \leq \frac{q_p + \sigma}{2^{n+\kappa}} + \frac{6 q_p \sigma}{2^{n+\kappa}} + \frac{\sigma^2}{2^{n+\kappa}} + \frac{2 q_v}{2^n},$$

where $\sigma = \sigma_e + \sigma_d + \sigma_v$.

Proof. First, we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{naead}^\star}_{\mathsf{LOCUS}[\widetilde{\mathsf{E}}]}(\mathscr{A}) &= \mathbf{Adv}^{\mathsf{naead}^\star}_{\Theta\text{-}\mathsf{LOCUS}[\widetilde{\mathsf{P}}[\widetilde{\mathsf{E}}]]}(\mathscr{A}) \\ &\leq \mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{\mathsf{P}}[\widetilde{\mathsf{E}}]}(\mathscr{A}) + \mathbf{Adv}^{\mathsf{naead}^\star}_{\Theta\text{-}\mathsf{LOCUS}[\widetilde{\Pi}[\widetilde{\mathsf{E}}]]}(\mathscr{A}) \\ &= \mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{\mathsf{P}}[\widetilde{\mathsf{E}}]}(\mathscr{A}) + \mathbf{Adv}^{\mathsf{naead}^\star}_{\Theta\text{-}\mathsf{LOCUS}[\widetilde{\Pi}]}(\mathscr{A}).\end{aligned}$$

The first equality holds trivially, as the Θ-$\mathsf{LOCUS}[\widetilde{\mathsf{P}}[\widetilde{\mathsf{E}}]]$ is just another view for $\mathsf{LOCUS}[\widetilde{\mathsf{E}}]$. The second inequality follows from a simple hybrid argument. Since, $\widetilde{\Pi}$ is independent of $\widetilde{\mathsf{E}}$, the NAEAD* advantage of $\mathscr{A}$ does not change if we drop $\widetilde{\mathsf{E}}$, whence the third equality.

In theorem 5, the TSPRP advantage of $\widetilde{\mathsf{P}}[\widetilde{\mathsf{E}}]$ is shown to be $O\left(\sigma^2/2^{n+\kappa} + q_p\sigma/2^{n+\kappa}\right)$, where $\sigma = \sigma_e + \sigma_d + \sigma_v$, and in theorem 2, the NAEAD* advantage of Θ-$\mathsf{LOCUS}[\widetilde{\Pi}]$ is shown to be $O(q_v/2^n)$. The result follows combining everything together. □

Theorem 2. *For any nonce-respecting $(q_e, q_d, q_v, \sigma_e, \sigma_d, \sigma_v)$-adversary $\mathscr{B}$, we have*

$$\mathbf{Adv}^{\mathsf{naead}^\star}_{\Theta\text{-}\mathsf{LOCUS}[\widetilde{\Pi}]}(\mathscr{B}) \leq \frac{2q_v}{2^n}.$$

Proof. We employ the coefficient-H technique. Let q denote the total number of construction queries made by $\mathscr{B}$, i.e., $q = q_e + q_d + q_v$. Further, let $[q]_e$, $[q]_d$, and $[q]_v$ denote the subset of encryption, decryption, and verification, respectively, query indices, i.e., $|[q]_x| = q_x$ for $x \in \{e, d, v\}$.

All the encryption query variables (including the internal ones) are defined analogous to algorithm 3 and 4. The variables arising in decryption and verification query are defined identically, but topped with tilde and bar, respectively, to differentiate them from their encryption counterparts.

Let denote the set of attainable transcripts in the ideal world. For any transcript $\omega \in$, we segregate the encryption, decryption, and verification query tuples into ω^e, ω^d, and ω^v, i.e. $\omega^e = (N^i, A^i, M^i, C^i, T^i)_{i\in[q]_e}$, $\omega^d = (\widetilde{N}^i, A^i, \widetilde{C}^i, \widetilde{M}^i)_{i\in[q]_d}$, $\omega^v = (\bar{N},\bar{A}^i, \bar{C}^i, \bar{T}^i, \bot^i)_{i\in[q]_v}$, and $\omega = \omega^e + \omega^d + \omega^v$.

We take all attainable transcripts to be good, i.e., $_{\mathsf{bad}} = \emptyset$. Now, for a good transcript ω, we claim that $\Pr[\Lambda_1^e = \omega^e, \Lambda_1^d = \omega^d] = \Pr[\Lambda_0^e = \omega^e, \Lambda_0^d = \omega^d]$. This can be easily argued due to our definition of $\$_{\mathsf{e}}$ and $\$_{\mathsf{d}}$, and the fact that the adversary only makes nonce-respecting encryption queries. Then, the ratio of interpolation probabilities is given by

$$\begin{aligned}\frac{\Pr[\Lambda_1 = \omega]}{\Pr[\Lambda_0 = \omega]} &= \Pr[\Lambda_1^v = \omega^v | \Lambda_1^e = \omega^e, \Lambda_1^d = \omega^d] \\ &\geq \left(1 - \Pr[\Lambda_1^v \neq \omega^v | \Lambda_1^e = \omega^e, \Lambda_1^d = \omega^d]\right),\end{aligned}$$

where we use the fact that $\Pr[\Lambda_0^v = \omega^v | \Lambda_0^e = \omega^e, \Lambda_0^d = \omega^d] = 1$ For $i \in [q]_v$, let $\mathtt{Forge}_i$ denote the event $(\bar{\mathsf{N}}^i, \bar{\mathsf{A}}^i, \bar{\mathsf{C}}^i, \bar{\mathsf{T}}^i, \bar{\lambda}^i) \neq (N^i, A^i, C^i, T^i, \bot) \mid \Lambda_1^e = \omega^e, \Lambda_1^d = \omega^d$, where $\bar{\lambda}^i$ denotes the output of the verification interface for the i-th verification query in the real oracle. Apart from $\bar{\lambda}^i$, all other variables are adversarial inputs, and hence must match. Then, we have

$$\Pr[\Lambda_1^v \neq \omega^v | \Lambda_1^e = \omega^e, \Lambda_1^d = \omega^d] \leq \sum_{i\in[q]_v} \Pr[\mathtt{Forge}_i].$$

We fix a verification query index i and follow the following two cases.

1. $\bar{\mathsf{N}}^i \neq \mathsf{N}^j$ for all $j \in [q]_e$. This means that in the real world, the tweakable random permutation $\widetilde{\Pi}$ was never called for tweak input $(\bar{\mathsf{N}}^i, 6, \cdot)$, whence the tag matches with at most 2^{-n} probability.
2. $\bar{\mathsf{N}}^i = \mathsf{N}^j$ for some $j \in [q]_e$. If $\bar{\mathsf{T}}^i \neq \mathsf{T}^j$, then the forgery succeeds with at most $1/(2^n - 1)$ probability, as this is equivalent of guessing the output of a uniform random permutation when one input-output pair is already known. Suppose $\bar{\mathsf{T}}^i = \mathsf{T}^j$. Then $(\bar{\mathsf{A}}^i, \bar{\mathsf{C}}^i) \neq (\mathsf{A}^j, \mathsf{C}^j)$, otherwise the queries are duplicate. Without loss of generality, we assume that $\bar{\mathsf{A}}^i = \mathsf{A}^j$. Then, there must be at least one ciphertext block index, say k, in $[\max\{|\bar{\mathsf{C}}^i|, \mathsf{C}^j\}]$ such that $\bar{\mathsf{C}}^i_k \neq \mathsf{C}^j_k$. Now, we have two cases based on $|\bar{\mathsf{C}}^i|$ and $|\mathsf{C}^j|$.
   a. $|\bar{\mathsf{C}}^i| \neq |\mathsf{C}^j|$, say $|\bar{\mathsf{C}}^i| > |\mathsf{C}^j|$. Then, we choose $k = \bar{\ell}_i$. In this case, we condition on the values of $\bar{\mathsf{W}}^i$ and W^j as well as $\bar{\mathsf{V}}^i$ and V^j, except $\bar{\mathsf{W}}^i_{\bar{\ell}_i}$. Then, the probability that $\bar{\mathsf{X}}^i_\oplus = \mathsf{X}^j_\oplus$ is bounded by at most $2/2^n$ due to the randomness of $\bar{\mathsf{W}}^i_{\bar{\ell}_i}$.
   b. $|\bar{\mathsf{C}}^i_k| = |\mathsf{C}^j_k|$. Suppose, the two ciphertexts differ only at the last block. Then it is easy to see that the probability of $\bar{\mathsf{X}}^i_\oplus = \mathsf{X}^j_\oplus$ is 0. This happens by design. Instead, suppose there exist $k < \ell_j$, such that $\bar{\mathsf{C}}^i_k \neq \mathsf{C}^j_k$. Then, the probability of $\bar{\mathsf{X}}^i_\oplus = \mathsf{X}^j_\oplus$ is bounded by $2/2^n$, using a similar line of argument as in the preceding case.

The cases 1, 2a, and 2b are all mutually exclusive, whence we can bound $\Pr[\mathtt{Forge}_i] \leq 2/2^n$. The result follows combining everything together. □

3.4 NAEAD* Security of LOTUS

In case of LOTUS, we again redefine $(\$_{\mathsf{e}}, \$_{\mathsf{d}})$ to make them similar to $(\Theta\text{-}\mathsf{LOTUS.enc}, \Theta\text{-}\mathsf{LOTUS.dec})$. Formally, we define $\$_{\mathsf{e}}$ as follows:

1. Initialize $\widetilde{\Pi} \quad {}_{\$}\,\mathrm{TPerms}((\mathcal{N}\times\mathbb{N}\times(3]), \{0,1\}^n)$ and $\Gamma \quad {}_{\$}\,\mathrm{Funcs}(\mathcal{N}\times\mathcal{A}\times\mathcal{M}, \{0,1\}^n)$, where $\mathrm{Funcs}(\mathcal{N}\times\mathcal{A}\times\mathcal{M}, \{0,1\}^n)$ denotes the set of all functions from $\mathcal{N}\times\mathcal{A}\times\mathcal{M}$ to $\{0,1\}^n$.

2. On input (N, A, M) do:

```
$a \quad \lceil |A|/n \rceil$, $d \quad \lceil |M|/2n \rceil$, $\ell \quad \lceil |M|/n \rceil$.
$(M_1, \ldots, M_\ell) \ {}^{n} \ M$.
For all $i \in [d-1]$,
$C_{2i-1} \quad \widetilde{\Pi}^{(N,a+i,0)}(M_{2i-1}) \oplus M_{2i}$.
$C_{2i} \quad \widetilde{\Pi}^{(N,a+i,1)}(C_{2i-1}) \oplus M_{2i-1}$.
End For
If $2d = \ell$ then,
$C_{2d-1} \quad \widetilde{\Pi}^{(N,a+d,2)}(\langle |M| - 2(d-1)n \rangle_n \oplus M_{2d-1}$.
$C_{2d} \quad \lfloor \widetilde{\Pi}^{(N,a+d,3)}(C_{2d}) \oplus \langle |M| - 2(d-1)n \rangle_n \rfloor_{|M_{2d}|} \oplus M_\ell$.
Else,
$C_\ell \quad \lfloor \widetilde{\Pi}^{(N,a+d,2)}(\langle |M| - 2(d-1)n \rangle_n) \rfloor_{|M_\ell|} \oplus M_\ell$.
End If
$C \quad C_1 \| \cdots \| C_\ell$.
$T \quad \Gamma(N, A, M)$.
Return $(C, T)$.
```

Again for nonce-respecting adversary the modified definition of $\$_{\mathsf{e}}$ is identical to the usual random string definition. This can be argued in a similar fashion as before. We define $\$_{\mathsf{d}}$ as the inverse of the redefined $\$_{\mathsf{e}}$.

The main technical result on the security of LOTUS is given in Theorem 3.

Theorem 3. *For any nonce-respecting $(q_e, q_d, q_v, q_p, \sigma_e, \sigma_d, \sigma_v)$-adversary $\mathscr{A}$, we have*

$$\mathbf{Adv}^{\mathsf{naead}^\star}_{\mathsf{LOTUS}[\widetilde{\mathsf{E}}]}(\mathscr{A}) \leq \frac{q_p + \sigma}{2^{n+\kappa}} + \frac{6 q_p \sigma}{2^{n+\kappa}} + \frac{\sigma^2}{2^{n+\kappa}} + \frac{2 q_v}{2^n},$$

where $\sigma = \sigma_e + \sigma_d + \sigma_v$.

Proof. We have

$$\begin{aligned} \mathbf{Adv}^{\mathsf{naead}^\star}_{\mathsf{LOTUS}[\widetilde{\mathsf{E}}]}(\mathscr{A}) &= \mathbf{Adv}^{\mathsf{naead}^\star}_{\Theta\text{-}\mathsf{LOTUS}[\widetilde{\mathsf{P}}[\widetilde{\mathsf{E}}]]}(\mathscr{A}) \\ &\leq \mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{\mathsf{P}}[\widetilde{\mathsf{E}}]}(\mathscr{A}) + \mathbf{Adv}^{\mathsf{naead}^\star}_{\Theta\text{-}\mathsf{LOTUS}[\widetilde{\Pi}[\widetilde{\mathsf{E}}]]}(\mathscr{A}) \\ &= \mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{\mathsf{P}}[\widetilde{\mathsf{E}}]}(\mathscr{A}) + \mathbf{Adv}^{\mathsf{naead}^\star}_{\Theta\text{-}\mathsf{LOTUS}[\widetilde{\Pi}]}(\mathscr{A}). \end{aligned}$$

The result follows from theorem 5 and 4. □

Theorem 4. *For any nonce-respecting $(q_e, q_d, q_v, \sigma_e, \sigma_d, \sigma_v)$-adversary $\mathscr{B}$, we have*

$$\mathbf{Adv}^{\mathsf{naead}^\star}_{\Theta\text{-}\mathsf{LOTUS}[\widetilde{\Pi}]}(\mathscr{B}) \leq \frac{2 q_v}{2^n}.$$

Proof. Given the renewed definition of $(\$_{\mathsf{e}}, \$_{\mathsf{d}})$, this proof is identical to the proof of theorem 2. □

3.5 Security of $\widetilde{\mathsf{P}}$

The main technical result on the security of $\widetilde{\mathsf{P}}$, as defined in section 3.1, is given in Theorem 5.

Theorem 5. *For any (q_e, q_d, q_p)-adversary $\mathscr{B}$, we have*

$$\mathbf{Adv}^{\mathsf{tsprp}}_{\widetilde{\mathsf{P}}[\widetilde{\mathsf{E}}]}(\mathscr{B}) \leq \frac{q_p + q}{2^{n+\kappa}} + \frac{6 q_p q}{2^{n+\kappa}} + \frac{q^2}{2^{n+\kappa}}.$$

Proof. We employ the coefficient-H technique to bound the distinguishing advantage of $\mathscr{B}$ in distinguishing the real oracle $(\widetilde{\mathsf{P}}^{\pm}, \widetilde{\mathsf{E}}^{\pm})$ from the ideal oracle $(\widetilde{\Pi}^{\pm}, \widetilde{\mathsf{E}}^{\pm})$. Let $[q]$ denote set of all construction query indices, and $[q]_e$, and $[q]_d$ denote the subset of encryption, and decryption, respectively, query indices, i.e., $|[q]_x| = q_x$ for $x \in \{e, d\}$.

For the i-th construction query, we define the following notations:

- $T_i := (N_i, d_i, m_i)$: the i-th tweak value; M_i: the i-th message; C_i: the i-th ciphertext.
- $K^i_N = K \oplus N$; $L_i := 2^{m_i} K^i_N$; $\Delta^i_N := \widetilde{\mathsf{E}}^1_{K^i_N}(\Delta_0)$, where $\Delta_0 = \widetilde{\mathsf{E}}^0_K(0)$.
- $X_i = M_i \oplus \Delta^i_N$; $Y_i = C_i \oplus \Delta^i_N$.

The i-th primitive query variables are defined analogously, but topped with a hat to differentiate them from their construction counterpart. So, the i-th primitive query is of the form $(\hat{L}_i, \hat{X}_i, \hat{Y}_i)$, where $\hat{L}_i$, $\hat{X}_i$, and $\hat{Y}_i$ denote the key, input and output of the primitive.

We consider an extended version of the oracles, in which they release the internal secrets, once the query-response phase is over. The real oracle releases the secret key K, and the Δ^i_{N} values for all $i \in [q]$. This uniquely defines all the intermediate variables arising in the construction queries.

The ideal oracle first samples a dummy key K uniformly at random. Let $\mathcal{S} = \{i \in [q] : \nexists j < i, \mathsf{N}_i = \mathsf{N}_j\}$. The ideal oracle samples Δ^i_{N} uniformly at random for all $i \in \mathcal{S}$, and sets $\Delta^j_{\mathsf{N}} = \Delta^i_{\mathsf{N}}$ if $\mathsf{N}_j = \mathsf{N}_i$ for all $j \in [q]$ and $i \in \mathcal{S}$. All other internal variables are defined according to their relationship in the real world.

Let denote the set of attainable transcripts in the ideal world. For any transcript $\omega \in$, we segregate the construction and primitive query tuples into ω^c, and ω^p, i.e. $\omega^c = (N_i, d_i, m_i, M_i, C_i, X_i, Y_i, \Delta^i_N, K^i_N, L_i)_{i \in [q]}$, $\omega^p = (\hat{L}_i, \hat{X}_i, \hat{Y}_i)_{i \in [q]_p}$.

Bad Transcript Analysis: We say that an attainable transcript is bad, if one of the following conditions hold:

C_0 : $\exists i \in [q]_p$ such that $K = \hat{L}_i$.

C_1 : $\exists i \in [q]$ such that $K = N_i$.

C_2 : $\exists i \in [q], j \in [q]_p$ such that $(\hat{L}_j, \hat{d}_j, \hat{Z}_j) = (K^i_N, 1, Z_i)$, where $(\hat{Z}_j, Z_i) \in \{(\hat{X}_j, \Delta_0), (\hat{Y}_j, \Delta^i_N)\}$.

C_3 : $\exists i \neq j \in [q]$ such that $(L_i, d_i, Z_i) = (L_j, d_j, Z_j)$, where $Z \in \{X, Y\}$.

C_4 : $\exists i \in [q], j \in [q]_p$ such that $(L_i, d_i, Z_i) = (\hat{L}_j, \hat{d}_j, \hat{Z}_j)$, where $Z \in \{X, Y\}$.

C_5 : $\exists i \in [q]$ such that $|\{j \in [q]_p : (\hat{L}_j, \hat{d}_j) = (L_i, d_i)\}| \geq 2^{n-1}$.

Let $\mathtt{bad}$ denote the event that Λ_0 satisfies one of the C_i for $i \in (5]$. Then, we have

$$\Pr[\Lambda_0 \in {}_{\mathtt{bad}}] = \Pr[\mathtt{bad}] = \Pr[\bigcup_{i=0}^{5} \mathsf{C}_i]. \tag{4}$$

It is easy to see that the probabilities $\Pr[\mathsf{C}_0]$ and $\Pr[\mathsf{C}_1]$ are bounded by at most $q_p 2^{-\kappa}$, and $q 2^{-\kappa}$, respectively, since K is chosen uniformly at random. Now, we bound the probabilities of C_2, $\mathsf{C}_3|\neg\mathsf{C}_1$, and C_4.

1. Bounding $\Pr[\mathsf{C}_2]$: In the ideal world, $\mathsf{K}^i_{\mathsf{N}}$, Δ_0, and Δ^i_{N} are all uniform and independent of each other. Further, there are two choices for $(\widehat{\mathsf{Z}}_j, \mathsf{Z}_i)$ and qq_p many choices for i and j. So the probability of this event can be bounded by at most $qq_p 2^{1-n-\kappa}$.

2. Bounding $\Pr[\mathsf{C}_3|\neg\mathsf{C}_1]$: Now we may have two cases:

1. $\mathsf{N}_i = \mathsf{N}_j$. In this case, we must have $\mathsf{m}_i = \mathsf{m}_j$, otherwise $\mathsf{L}_i = 2^{\mathsf{m}_i}\mathsf{K}^i_{\mathsf{N}} \neq 2^{\mathsf{m}_j}\mathsf{K}^i_{\mathsf{N}} = \mathsf{L}_j$. Now $\mathsf{X}_i = \mathsf{X}_j$ implies that $\mathsf{M}_i = \mathsf{M}_j$ and $\mathsf{X}_i = \mathsf{X}_j$ implies that $\mathsf{C}_i = \mathsf{C}_j$, both of which imply duplicate query. So the probability is zero in this case.
2. $\mathsf{N}_i \neq \mathsf{N}_j$. In this case, we have two equations $2^{\mathsf{m}_i}\mathsf{K}^i_{\mathsf{N}} = 2^{\mathsf{m}_j}\mathsf{K}^j_{\mathsf{N}}$ and $\mathsf{Z}_i = \mathsf{Z}_j$ in two independent random variables (K, and Δ^i_{N}) which gives a probability of $2^{-n-\kappa}$. Further, there are 2 choices for Z and $\binom{q}{2}$ choices for i and j. Thus, we have $\Pr[\mathsf{C}_3|\neg\mathsf{C}_1] \leq q^2 2^{-n-\kappa}$.

3. Bounding $\Pr[\mathsf{C}_4]$: This event is similar to 2. above, and the probability can be bounded by at most $qq_p 2^{1-n-\kappa}$ using the randomness of K and Δ^i_{N}.

4. Bounding $\Pr[\mathsf{C}_5]$: This event is mainly useful in avoiding the case when the adversary accidentally exhausts the entire codebook for some construction query key L_i. Let $\widehat{\mathcal{K}}$ denote the set of all indices $i \in [q_p]$ such that $|\{j \in [q_p] : j > i, \widehat{\mathsf{L}}_j = \widehat{\mathsf{L}}_i\}| \geq 2^{n-1}$. Then $|\widehat{\mathcal{K}}| \leq q_p/2^{n-1}$. Since K is uniformly distributed, we have

$$\Pr[\mathsf{C}_5] = \Pr[\mathsf{L}_i \in \widehat{\mathcal{K}}] \leq \frac{2q_p q}{2^{n+\kappa}}.$$

On combining all bounds, we get

$$\Pr[\mathsf{bad}] \leq \frac{q_p + q}{2^{n+\kappa}} + \frac{6 q_p q}{2^{n+\kappa}} + \frac{q^2}{2^{n+\kappa}}.$$

Good Transcript Analysis: Fix any good transcript ω. Let $(T'_1, \ldots, T'_r)$ denote the distinct tweaks present in $(T_1, \ldots, T_q)$. Let $(c_1, \ldots, c_r)$ be a tuple of positive integers with $c_i = |\{j \in [q] : T_j = T'_i\}|$. Clearly, $\sum_j c_j = q$ since the transcript is good (i.e. $(T_i, X_i) = (T_j, X_j) \iff (T_i, Y_i) = (T_j, Y_j)$). Now, in the ideal world we have

$$\begin{aligned}\Pr[\Lambda_0 = \omega] &= \Pr[\Lambda_0^p = \omega^p, \Lambda_0^c = \omega^c] \\ &= \Pr[\Lambda_0^p = \omega^p] \cdot \frac{1}{2^\kappa 2^{n(q+1)} \prod_{i=1}^{r} (2^n)_{c_i}} \qquad (5)\end{aligned}$$

Let $((L'_1, d'_1), \ldots, (L'_s, d'_s))$ denote the distinct keys and short tweak tuples present in $((L_1, d_1), \ldots, (L_q, d_q))$. Let $(a_1, \ldots, a_s)$ and $(b_1, \ldots, b_s)$ be tuples of positive integers such that $a_i = |\{j \in [q] : (L_j, d_j) = (L'_i, d'_i)\}|$ and $b_i = |\{j \in [q]_p : (\hat{L}_j, \hat{d}_j) = (L'_i, d'_i)\}|$. Clearly, $\sum_{i=1}^{s} a_i = q$, and $b_i < 2^{n-1}$ for all $i \in [s]$, since the transcript is good. Then, we have

$$\begin{aligned}\Pr[\Lambda_1 = \omega] &= \Pr[\Lambda_1^p = \omega^p] \cdot \Pr[\Lambda_1^c = \omega^c | \Lambda_1^p = \omega^p] \\ &= \Pr[\Lambda_0^p = \omega^p] \cdot \frac{1}{2^\kappa 2^{n(q+1)} \prod_{i=1}^{s} (2^n - b_i)_{a_i}} \qquad (6)\end{aligned}$$

On dividing Eq. (6) by (5), and doing some simple algebraic simplifications, we get

$$\frac{\Pr[\Lambda_1 = \omega]}{\Pr[\Lambda_0 = \omega]} \geq 1.$$

The result follows from coefficient-H technique. □

References

[1] Elena Andreeva, Begül Bilgin, Andrey Bogdanov, Atul Luykx, Florian Mendel, Bart Mennink, Nicky Mouha, Qingju Wang, and Kan Yasuda. PRIMATEs v1.02. Submission to CAESAR, 2016. https://competitions.cr.yp.to/round2/primatesv102.pdf.

[2] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Elmar Tischhauser, and Kan Yasuda. AES-COPA v.2. Submission to CAESAR, 2015. https://competitions.cr.yp.to/round2/aescopav2.pdf.

[3] ATHENa: Automated Tool for Hardware Evaluation. https://cryptography.gmu.edu/athena.

[4] Authenticated Encryption FPGA Ranking. https://cryptography.gmu.edu/athenadb/fpga_auth_cipher/rankings_view.

[5] Jean-Philippe Aumasson, Philipp Jovanovic, and Samuel Neves. NORX v3.0. Submission to CAESAR, 2016. https://competitions.cr.yp.to/round3/norxv30.pdf.

[6] Subhadeep Banik, Andrey Bogdanov, Atul Luykx, and Elmar Tischhauser. Sundae: Small universal deterministic authenticated encryption for the internet of things. *IACR Transactions on Symmetric Cryptology*, 2018(3):1–35, Sep. 2018.

[7] Guido Bertoni, Michaël Peeters Joan Daemen, Gilles Van Assche, and Ronny Van Keer. Ketje v2. Submission to CAESAR, 2016. https://competitions.cr.yp.to/round3/ketjev2.pdf.

[8] Avik Chakraborti, Anupam Chattopadhyay, Muhammad Hassan, and Mridul Nandi. Trivia: A fast and secure authenticated encryption scheme. In *CHES 2015*, pages 330–353, 2015.

[9] Avik Chakraborti, Anupam Chattopadhyay, Muhammad Hassan, and Mridul Nandi. Trivia and utrivia: two fast and secure authenticated encryption schemes. *J. Cryptographic Engineering*, 8(1):29–48, 2018.

[10] Avik Chakraborti, Nilanjan Datta, Ashwin Jha, Cuauhtemoc Mancillas Lopez, Mridul Nandi, and Yu Sasaki. LOTUS-AEAD and LOCUS-AEAD. Submission to NIST Lightweight Competition, 2019. https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/lotus-aead-and-locus-aead-spec.pdf.

[11] Avik Chakraborti, Nilanjan Datta, Ashwin Jha, Cuauhtemoc Mancilias-López, Mridul Nandi, and Yu Sasaki. LOTUS-AEAD and LOCUS-AEAD. Submission to NIST LwC Standardization Process (Round 1), 2019.

[12] Avik Chakraborti, Nilanjan Datta, Mridul Nandi, and Kan Yasuda. Beetle family of lightweight and secure authenticated encryption ciphers. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):218–241, 2018.

[13] Avik Chakraborti, Tetsu Iwata, Kazuhiko Minematsu, and Mridul Nandi. Blockcipher-based authenticated encryption: How small can we go? In *CHES 2017*, pages 277–298, 2017.

[14] Avik Chakraborti and Mridul Nandi. TriviA-ck-v2. Submission to CAESAR, 2015. https://competitions.cr.yp.to/round2/triviackv2.pdf.

[15] Nilanjan Datta and Mridul Nandi. Proposal of ELmD v2.1. Submission to CAESAR, 2015. https://competitions.cr.yp.to/round2/elmdv21.pdf.

[16] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2. Submission to CAESAR, 2016. https://competitions.cr.yp.to/round3/asconv12.pdf.

[17] Morris Dworkin. Recommendation for block cipher modes of operation: Galois/counter mode (GCM) and GMAC, NIST Special Publication 800-38D, 2011. csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf.

[18] Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, Sumio Morioka, and Eita Kobayashi. CLOC and SILC. Submission to CAESAR, 2016. https://competitions.cr.yp.to/round3/clocsilcv3.pdf.

[19] Jérémy Jean, Ivica Nikolić, and Thomas Peyrin. Deoxys v1.41. Submission to CAESAR, 2016. https://competitions.cr.yp.to/round3/deoxysv141.pdf.

[20] B. Jungk and M. Stttinger. Hobbit: Smaller but faster than a dwarf: Revisiting lightweight SHA-3 FPGA implementations. In *2016 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, pages 1–7, 2016.

[21] Ted Krovetz and Phillip Rogaway. OCB(v1.1). Submission to CAESAR, 2016. https://competitions.cr.yp.to/round3/ocbv11.pdf.

[22] Sachin Kumar, Jawad Haj-Yihia, Mustafa Khairallah, and Anupam Chattopadhyay. A comprehensive performance analysis of hardware implementations of CAESAR candidates. *IACR Cryptology ePrint Archive*, 2017:1261, 2017.

[23] Kazuhiko Minematsu. AES-OTR v3.1. Submission to CAESAR, 2016. https://competitions.cr.yp.to/round3/aesotrv31.pdf.

[24] Yusuke Naito, Mitsuru Matsui, Takeshi Sugawara, and Daisuke Suzuki. SAEB: A lightweight blockcipher-based AEAD mode of operation. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):192–217, 2018.

[25] Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, pages 16–31, 2004.

[26] Hongjun Wu. ACORN: A Lightweight Authenticated Cipher (v3). Submission to CAESAR, 2016. https://competitions.cr.yp.to/round3/acornv3.pdf.

[27] Hongjun Wu and Tao Huang. The JAMBU Lightweight Authentication Encryption Mode (v2.1). Submission to CAESAR, 2016. https://competitions.cr.yp.to/round3/jambuv21.pdf.

[28] Hongjun Wu and Bart Preneel. AEGIS : A Fast Authenticated Encryption Algorithm (v1.1). Submission to CAESAR, 2016. https://competitions.cr.yp.to/round3/aegisv11.pdf.