ORANGE

Designers/Submitters:
Bishwajit Chakraborty, Mridul Nandi
Indian Statistical Institute, Kolkata
bishu.math.ynwa@gmail.com, mridul.nandi@gmail.com

September 20, 2019

1 Introduction

This work proposes ORANGE, a variant of sponge authenticated encryption and sponge hash which can absorb data in the optimum rate. In other words, it is an Optimum RAte spoNGE construction. In this submission, we propose an authenticated encryption, named as ORANGE-Zest and a hash function, named as ORANGISH based on a 256-bit permutation.

Underlying Permutation. Both construction use PHOTON_{256} as the underlying permutation. Among the existing 256-bit permutations, PHOTON_{256} [6] is one of the lightest designs in the literature. It has been well studied and well analysized. Moreover, PHOTON_{256} is also a part of ISO-IEC: 29192-5 standard, which deal specifically with light-weight cryptography.

Hash Mode. The mode of hash function ORANGISH is very close to the JH hash function [13] which is one of the finalists of SHA3-competition. JH mode allows us to absorb 128 bit data for each permutation call. Thus, it has higher throughput compared with classical sponge hash function [1, 2]. The design of ORANGISH is expected to provide collision and preimage security against all adversaries running in time 2^{112} (i.e. making 2^{112} permutation calls).

Authenticated Encryption Mode. The mode for ORANGE-Zest is a close variant of sponge with full state absorption. The full state absorption is possible as we hold another state of size 128-bits, a part of the output of previous execution of the underlying permutation. We use this dynamic secret state to mask a part of the ciphertext. This mode can be easily generalized to a design based on a permutation with $2n$ bit state. In our case, $n = 128$. To summarize the performance of our AE mode, it has $3n$ bit state with $2n$ bit rate. To process $2n$ bit blocks, we apply $4n$-bit XOR, in addition to one permutation call. The design of ORANGE is expected to provide privacy and confidentiality against all adversaries running in time 2^{128} (i.e. making 2^{128} permutation calls) having at most 2^{64} data.

2 Notations and Conventions

We use $\{0,1\}^+$ and $\{0,1\}^n$ to denote the set of all non-empty (binary) strings, and n-bit strings, respectively. λ denotes the empty string and $\{0,1\}^* = \{0,1\}^+ \cup \{\lambda\}$. For all practical purposes: we use little-endian format of indexing, and assume all binary strings are *byte-oriented*, i.e. belong in $(\{0,1\}^8)^*$. For any string $B \in \{0,1\}^+$, $|B|$ denotes the number of bits in B, and for $0 \le i \le |B|-1$, b_i denotes the i-th bit of B, i.e. $B = b_{|B|-1} \cdots b_0$. where b_0 is the least significant bit (LSB) and $b_{|B|-1}$ is the most significant bit (MSB). Given a nonempty bit string B of size $x < n$, we denote $\mathsf{pad}(B)$ as $0^{n-x-1}1B$. Thus we always pad the extra bits from MSB side. When $x = n$, we define $\mathsf{pad}(B)$ as B itself. The chop function chops either the most significant or least significant bits. For $k \le n$, and $B \in \{0,1\}^n$, $\lfloor B \rfloor_k := B_{k-1} \dots B_0$ and $\lceil B \rceil_k := B_{n-1} \dots B_{n-k}$.

For $B \in \{0,1\}^+$, $(B_{\ell-1}, \dots, B_0) \xleftarrow{n} B$, denotes the n-bit *block parsing* of B into $(B_{\ell-1}, \dots, B_0)$, where $|B_i| = n$ for $0 \le i \le \ell-2$, and $1 \le |B_{\ell-1}| \le n$. For $A, B \in \{0,1\}^+$, and $|A| = |B|$, $A \oplus B$ denotes the "bitwise XOR" operation on A and B. For $A, B \in \{0,1\}^+$, $A \| B$ denotes the "string concatenation" operation on A and B. For any $B \in \{0,1\}^+$ and a non-negative integer s, $B \ll s$ and $B \lll s$ denote the "left shift by s" and "circular left shift by s" operations on B, respectively. The notations for right shift and circular right shift are analogously defined using $\gg$ and $\ggg$, respectively. Given two matrices $M_{m\times l}$ and $N_{l\times n}$, $M \cdot N$ denotes the matrix multiplication of M and N.

We will use a compact representation of if-else statement by the following expression $P \; ? \; b \; : \; c$ where P is some mathematical statement. This evaluates to b if P is true and c otherwise. $P_1 \;\&\; P_2 \; ? \; b_1 \; : \; b_2 \; : \; b_3 \; : \; b_4$ evaluates to b_1 if both P_1 and P_2 are true, to b_2 if only P_1 is true, to b_3 if only P_2 is true and to b_4 if none of P_1 , P_2 are true.

Field Multiplication . It is well known that $x^{128} + x^7 + x^2 + x + 1$ is a primitive polynomial over the finite field of order 2. We define a constant $a := 0^{120}10000111$. Given $B \in \{0,1\}^{128}$, the α-multiplication on an 128 bit string $B := b_{127} \cdots b_1 b_0$, denoted by $\alpha \cdot B$, is defined as $(B \ll 1) \oplus a$ if $b_{127} = 1$, $B \ll 1$, otherwise. For a $c \in \mathbb{Z}_{\ge 0}$, $\alpha^c \cdot B$ denotes c times repeated α-multiplication of B.

2.1 Our Recommendation

ORANGE is primarily parameterized by its underlying Permutation P. We choose P to be PHOTON_{256} as described in Algorithm 2. We propose a hash function, called ORANGISH, and authenticated encryption ORANGE-Zest. Description of both are given in 1. Our proposal of ORANGE-Zest uses a nonce-size of 128−bits and a key-size of 128−bits to produce a 128-bit tag. It is clear from the description that the hash function ORANGISH is very close to the process of associated data in ORANGE-Zest. So a combined implementation of both ORANGISH and ORANGE-Zest would be optimized.

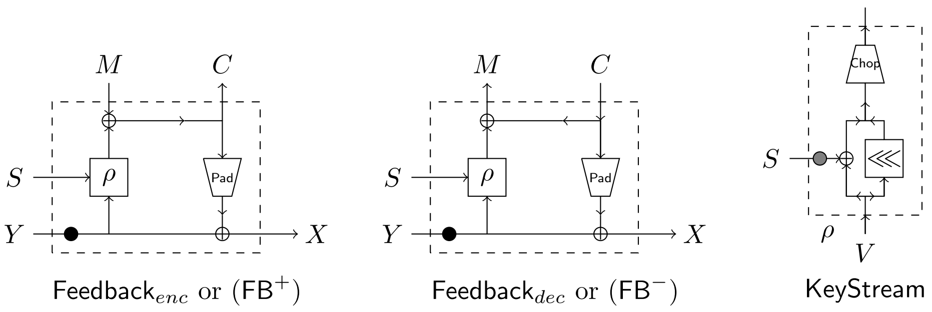

Figure 1: Feedback process for ORANGE-Zest: KeyStream module or the function ρ describes how the key-stream is defined. Feedback functions describe to define the next input X for the block cipher and the ciphertext (for encryption feedback) and message (for decryption feedback). The black circular dot represents the mult operation which is nothing but the α^{δ_M}-multiplication to the most significant half of Y (the previous block cipher output). Note that $\delta_M = 0, 1, 2$ for imtermediate block, complete last block, partial last block respectively .The gray circular dot represents the mult operation which is nothing but the α-multiplication to S. Here, Pad and Chop, pads and chops appropriate amounts of bits from MSB or LSB sides. The exact definitions of these process can be found in Algorithm 1

3 PHOTON$_{256}$ Permutation

We use PHOTON$_{256}$ [6] as our underlying 256-bit permutation in our mode. We use exactly same permutation without changing any part of the definition as it has been well studied. However, for the sake of completeness we provide a brief description of the permutation in this section (see Algorithm 2). It is applied on a state of 64 elements of 4 bits each, which is represented as a (8×8) matrix X. Let $X[i, j]$ denote the element at i-th row and j-th column of X.

PHOTON$_{256}$ is composed of 12 rounds. Each round applies four layers of functions AddConstant, SubCells, ShiftRows and MixColumnSerial on the state in a sequence. The description of these functions are given in Algorithm2. Informally, AddConstant adds fixed constants to the cells of the internal state. SubCells applies an 4-bit S-Box (see Table. 1) to each of the 64 4-bit cells. ShiftRows rotates the position of the cells in each of the rows and MixColumnSerial linearly mixes all the columns independently using a serial matrix multiplication. The multiplication with the coefficients in the matrix is in $GF(2^4)$ with $x^4 + x + 1$ being the irreducible polynomial.

We represent a serial matrix Serial$[a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7]$ by

$$\mathrm{Serial}[a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7] := \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \end{pmatrix}.$$

Table 1: The PHOTON S-box

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S-box	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Algorithm 1 ORANGE-Zest and ORANGISH and their main modules. Here, $\perp$ and $\top$ denote the abort and accept symbols respectively.

function ORANGE-ZEST$_{[\mathsf{P}]}$.enc(K, N, A, M)
  $(A_{a-1}, \ldots, A_0) \xleftarrow{n} A$
  $(M_{m-1}, \ldots, M_0) \xleftarrow{n} M$
  if $a = 0, m = 0$ **then**
    $(T, *) \leftarrow \mathsf{P}((K \oplus 2) \| N)$
    return (λ, T)
  if $a = 0, m \neq 0$ **then**
    $(C, U) \leftarrow \mathsf{proc\_txt}(K, (K \oplus 1) \| N, M, +)$
    return $(C, \mathsf{proc\_tg}(U))$
  $C \leftarrow \lambda$
  if $a \neq 0$ **then** $U \leftarrow \mathsf{proc\_hash}(K \| N, A, 1, 2)$
  if $a \neq 0, m \neq 0$ **then** $(C, U) \leftarrow \mathsf{proc\_txt}(K, U, M, +)$
  return $(C, \mathsf{proc\_tg}(U))$

function ORANGISH(D)
  $(D_{d-1}, \ldots, D_0) \xleftarrow{n} D$
  $D_d \leftarrow (n \nmid |D_{d-1}|)?\ 0^{n-2}10 : 0^{n-1}1$
  $D_{d-1} \leftarrow \mathsf{pad}(D_{d-1})$
  $X \leftarrow (0^n || D_0)$
  for $i = 0$ to $d-1$ **do**
    $A_i \leftarrow (D_i \| D_{i+1})$
  $Z \leftarrow \mathsf{proc\_hash}(X, (A_{d-1} \| \ldots \|, A_0), 0, 0)$
  $Z_1 \leftarrow \mathsf{P}(Z)$
  $Z_2 \leftarrow \mathsf{P}(Z_1)$
  return $\lfloor Z_2 \rfloor_n \| \lfloor Z_1 \rfloor_n$

function $\mathsf{proc\_txt}(S_0, U_0, D, \mathsf{dir})$
  $(D_{d-1}, \ldots, D_0) \xleftarrow{2n} D$
  for $i = 0$ **to** $d-1$ **do**
    $V_i \leftarrow \mathsf{P}(U_i)$
    if $i = d-1$ **then**
      $c \leftarrow (2n \mid |D_{d-1}|)?1 : 2$
      $V_i \leftarrow \mathsf{mult}(c, V_i)$
    $KS_i \leftarrow \rho(S_i, V_i)$
    $D_i' \leftarrow D_i \oplus \lfloor KS_i \rfloor_{|D_i|}$
    if $\mathsf{dir} = "+"$ **then** $D_i \leftarrow D_i'$
    $S_{i+1} \leftarrow \lceil V_i \rceil_n$
    $U_{i+1} \leftarrow V_i \oplus \mathsf{pad}(D_i)$
  return (D', U_d)

function ORANGE-ZEST$_{[\mathsf{P}]}$.dec(K, N, A, C, T)
  $(A_{a-1}, \ldots, A_0) \xleftarrow{n} A$
  $(C_{m-1}, \ldots, C_0) \xleftarrow{n} C,\ M \leftarrow \lambda$
  if $a = 0, m = 0$ **then** $(T', *) \leftarrow \mathsf{P}((K \oplus 2) \| N)$
  if $a = 0, m \neq 0$ **then**
    $(M, U) \leftarrow \mathsf{proc\_txt}(K, (K \oplus 1) \| N, C, -)$
    $T' \leftarrow \mathsf{proc\_tg}(U)$
  if $a \neq 0$ **then** $U \leftarrow \mathsf{proc\_hash}(N \| K, A, 1, 2)$
  if $a \neq 0, m \neq 0$ **then** $(M, U) \leftarrow \mathsf{proc\_txt}(K, U, C, -)$
  $T' \leftarrow \mathsf{proc\_tg}(U)$
  if $T \neq T'$ **then**
    return $\perp$
  else
    return $(M, \top)$

function $\mathsf{proc\_hash}(X, D, c_0, c_1)$
  $(D_{d-1}, \ldots, D_0) \xleftarrow{2n} D$
  $X_0 \leftarrow X$
  for $i = 0$ **to** $d-2$ **do**
    $Y_i \leftarrow \mathsf{P}(X_i)$
    $X_{i+1} \leftarrow Y_i \oplus D_i$
  $c \leftarrow (2n \mid |D_{d-1}|)?c_0 : c_1$
  $Y_{d-1} \leftarrow \mathsf{P}(X_{d-1})$
  $Y_{d-1} \leftarrow \mathsf{mult}(c, Y_{d-1})$
  $X_d \leftarrow Y_{d-1} \oplus \mathsf{pad}(D_{d-1})$
  return X_d

function $\rho(S, Y)$
  $(Y^b, Y^t) \xleftarrow{n} Y$
  $Z \leftarrow (Y^b \oplus \alpha S) \| (Y^t \lll 1)$
  return Z

function $\mathsf{mult}(c, V)$
  $(V^b, V^t) \xleftarrow{n} V$
  return $\alpha^c \cdot V^b \parallel V^t$

function $\mathsf{proc\_tg}(U)$
  $(U^b, U^t) \xleftarrow{n} U$
  return $\mathsf{P}(U^t \| U^b)$

4 Security of ORANGE

Here we describe some possible strategies to attack the ORANGE mode, and give a rough estimate on the amount of data and time required to mount those attacks (see Table 2). In the following discussion:

- D denotes the data complexity of the attack. This parameter quantifies the online resource requirements, and includes the total number of blocks (among all messages and associated data) processed through the underlying permutation for a fixed master key. Note that for simplicity we also use D to denote the data complexity of forging attempts.

- T denotes the time complexity of the attack. This parameter quantifies the offline resource requirements, and includes the total time required to process the off line evaluations of the underlying permutation. Since one call of the permutation can be assumed to take a constant amount of time, we generally take T as the total number of off line calls to the permutation.

Security Model	Data complexity ($\log_2 D$)	Time complexity ($\log_2 T$)
IND-CPA	64	128
INT-CTXT	64	128

Table 2: Security Claims of ORANGE-Zest. We remark that the given values indicate the amount of data or time required to make the attack advantage close to 1.

Algorithm 2 PHOTON$_{256}$ Modules. Note that we view the state X as a matrix and $M^8 \cdot X$ in MixColumnSerial represents the matrix multiplication in the underlying field $GF(2^4)$ defined over the irreducible polynomial $x^4 + x + 1$.

```
function AddConstant(X, K)
    RC[12] ← {1, 3, 7, 14, 13, 11, 6, 12, 9, 2, 5, 10}
    IC[8] ← {0, 1, 3, 7, 15, 14, 12, 8}
    for i = 0 to 7 do
        X[i, 0] ← X[i, 0] ⊕ RC[k] ⊕ IC[i]
    return X

function SubCells(X)
    for i = 0  to 7    j=0 to  7 do
        X[i, j] ← S-box(X[i, j])
    return X

function ShiftRows(X)
    for i = 0 to 7   j = 0 to 7 do
        X′[i, j] ← X[i, (j + i)%4]
    return X′

function MixColumnSerial(X)
    M ← Serial[2, 4, 2, 11, 2, 8, 5, 6]
    M^8 · X
    return X

function PHOTON_256(X)
    for i = 0 to 11 do
        X ← AddConstant(X)
        X ← SubCells(X)
        X ← ShiftRows(X)
        X ← MixColumnSerial(X)
    return X
```

Table 3: Security of Hash Function Family ORANGISH.

Mode	Security	Time complexity security (in bits)
ORANGISH	Collision	112
ORANGISH	Pre-image	128

4.1 IND-CPA and INT-CTXT Security of ORANGE-Zest

The privacy security of a permutation based construction relies on no collision of the inputs among online and offline permutation calls. Note that both top and bottom part of the input of a permutation call during the computation of an encryption query has full entropy due to two previous outputs. Hence a collision would happen with probability at most $1/2^{-256}$. The privacy claim of our design follows from this observation.

Note that the tag verification algorithm is almost same as that of Beetle [3]. Hence, a similar argument follows.

4.2 Collision Security of ORANGISH

To mount a collision attack on ORANGISH, suppose an adversary can make q many permutation calls. Suppose all the states reachable from the initial state (we define the initial state as 0^{256}) using the permutation calls are called *reachable states*. The adversary can set up the queries in an adaptive way to make all the query inputs (and hence query outputs) *reachable states*. We claim that the number of reachable state can be at most nq (by using multi-collision argument, details will be provided later). Hence, finding a collision pair has probability at most $n^2q^2/2^{256}$. This leads to our claim on the collision security.

4.3 Preimage Security of ORANGISH

In ORANGISH we set the tag size as 256 bits and the tag squeeze rate as 128 bits. So given a preimage target $T_2 \| T_1$, an adversary needs to find a Z such that $\mathsf{PHOTON}_{256}(Z\|T_1) = \star\|T_2$ or $\mathsf{PHOTON}_{256}^{-1}(Z\|T_2) = \star\|T_1$. It is easy to see that the probability of this event can be bounded by $\frac{q}{2^{128}}$ where q is the number of P and P^{-1} call.

5 Existing Analysis of PHOTON$_{256}$

Basic security analysis for PHOTON$_{256}$ has been provided explicitly in the original paper [6]. PHOTON is an ISO standard with a comfortable security margin. As we have used PHOTON$_{256}$ we only report briefly the known analysis of it.

A rebound-like attack [6] allows us to distinguish 8 rounds of PHOTON_{256} from an ideal permutation of the same size with time complexity 2^{16} (and later reduced to $2^{10.8}$[8]) and memory complexity of 2^8. In [7] Jean et al. presented a distinguisher for 9 round PHOTON_{256} with time complexity of 2^{184} and memory complexity of 2^{32}. Some other attacks are improved Indifferentiable [10] and statistical Integral distinguisher [5]. Recently, Wang et al. [12] presented the first full round distinguishers on PHOTON_{256} based on zero-sum partitions of size 2^{184}.

We believe that all these distinguishers have no impact on the security of our construction as these attacks are much more costlier than the security target we are aiming.

6 Design Rational

6.1 Choice of the Mode

Our primary goal is to design a lightweight cipher that has optimum throughput. No such sponge variant is known so far which can absorb message at the rate of the state of the permutation. Our design achieves this at the cost of an additional state. So it is optimum in rate. We also use JH variant of hash which also absorbs much higher data compared with classical sponge hash.

6.2 Need of an additional state

A b-bit permutation with r bit rate leaks r bit information about the permutation outputs. So when $r = b$, all the state value would be leaked and the key can be computed easily. Thus we need additional state to keep some amount of secret. We find that 128 bit additional state (chosen dynamically) provides the desired security.

6.3 Choice of the Permutation

PHOTON is an ISO-standard lightweight permutation which also provides sufficient amount of security level.

7 Figures of ORANGE for Different Cases

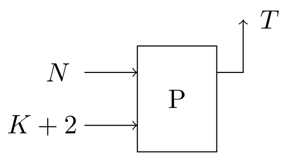

Figure 2: ORANGE-Zest encryption ($|A| = 0, |M| = 0$).

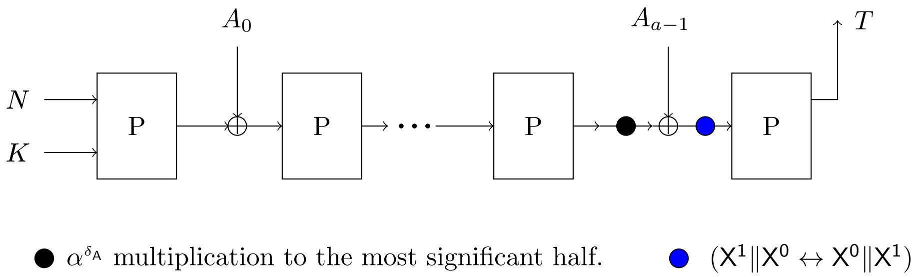

Figure 3: ORANGE-Zest encryption ($|M| = 0, |A| \neq 0, \delta_A = 1/2$ for complete-last/ partial block).

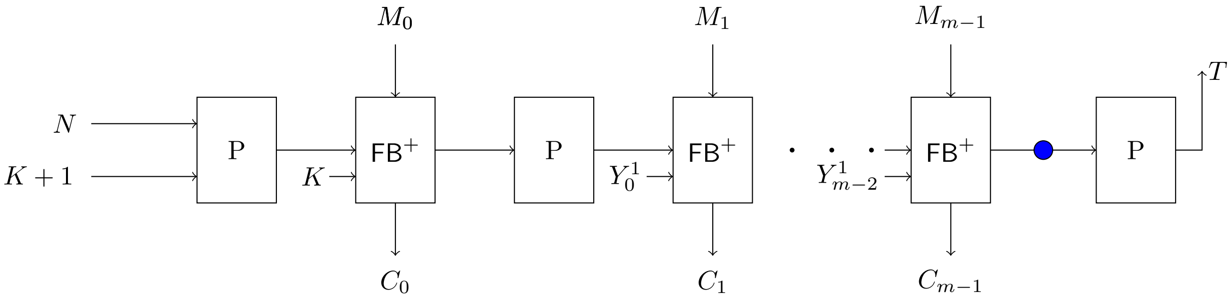

Figure 4: ORANGE-Zest encryption ($|A| = 0, |M| \neq 0$). Here $Y^1 = \lceil Y \rceil_{\frac{n}{2}}$.

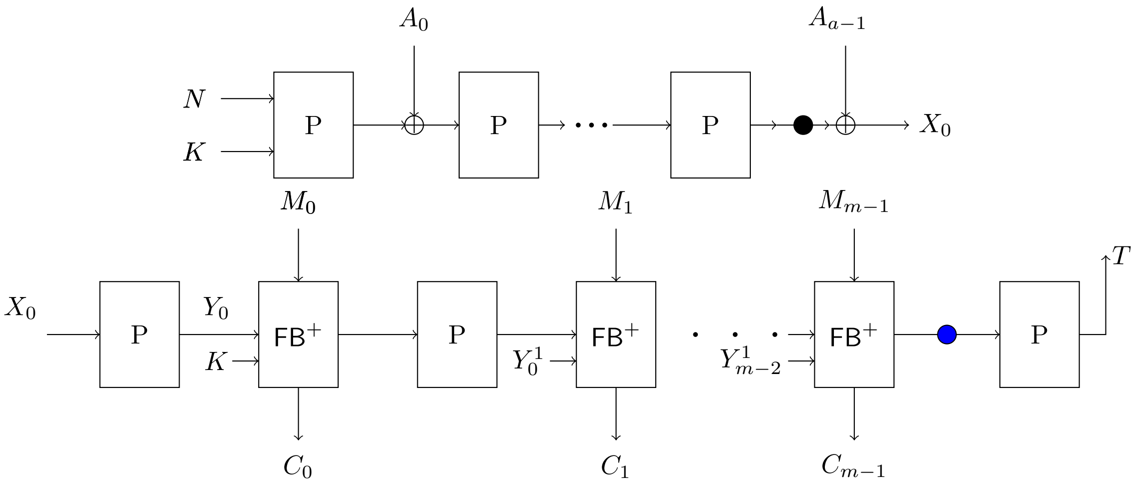

Figure 5: ORANGE-Zest encryption ($|A| \neq 0, |M| \neq 0$)

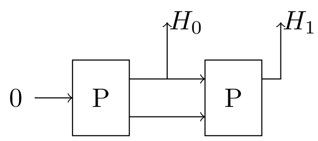

Figure 6: ORANGISH output ($|M| = 0$). The final hash output is defined as $H_1 \| H_0$.

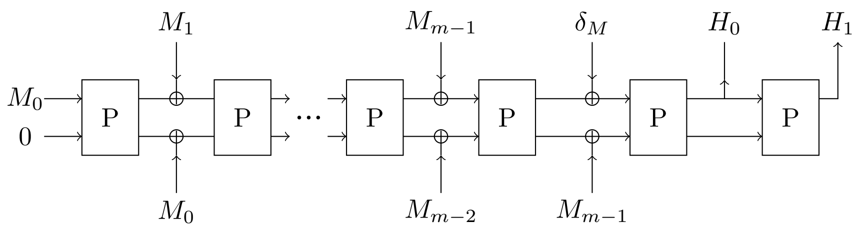

Figure 7: ORANGISH output ($|M| \neq 0$, $\delta_M = 1/2$ for complete/ partial input). The final hash output is defined as $H_1 \| H_0$.

8 Background for Proof of ORANGE-Zest

8.1 H-coefficient Technique

Consider a computationally unbounded and deterministic adversary $\mathscr{A}$ that tries to distinguish the real oracle, say $\mathcal{O}_1$, from the ideal oracle, say $\mathcal{O}_0$. We denote the query-response tuple of $\mathscr{A}$'s interaction with its oracle by a transcript ω. Sometimes, this may also include any additional information that the oracle chooses to reveal to the distinguisher at the end of the query-response phase of the game. We will consider this extended definition of transcript. We denote by Θ_1 (res. Θ_0) the random transcript variable when $\mathscr{A}$

interacts with $\mathcal{O}_1$ (res. $\mathcal{O}_0$). The probability of realizing a given transcript ω in the security game with an oracle $\mathcal{O}$ is known as the *interpolation probability* of ω with respect to $\mathcal{O}$. Since $\mathscr{A}$ is deterministic, this probability depends only on the oracle $\mathcal{O}$ and the transcript ω. A transcript ω is said to be *attainable* if $\Pr[\Theta_0 = \omega] > 0$. In this paper, $\mathcal{O}_1 = (\mathsf{enc}_\mathsf{K}, \mathsf{dec}_\mathsf{K}, f^{\pm})$, $\mathcal{O}_0 = (\Gamma, \perp, f^{\pm})$, and the adversary is trying to distinguish $\mathcal{O}_1$ from $\mathcal{O}_0$ in AEAD sense. Now we state a simple yet powerful tool due to Patarin [11], known as the H-coefficient technique (or simply the H-technique).

Theorem 1 (H-coefficient technique [11]). *Let Ω be the set of all realizable transcripts. For some $\epsilon_{\mathsf{bad}}, \epsilon_{\mathsf{ratio}} > 0$, suppose there is a set $\Omega_{\mathsf{bad}} \subseteq \Omega$ satisfying the following:*

- $\Pr[\Theta_0 \in \Omega_{\mathsf{bad}}] \leq \epsilon_{\mathsf{bad}}$;
- *For any $\omega \notin \Omega_{\mathsf{bad}}$,*
$$\frac{\Pr[\Theta_1 = \omega]}{\Pr[\Theta_0 = \omega]} \geq 1 - \epsilon_{\mathsf{ratio}}.$$

Then for any adversary $\mathscr{A}$, we have the following bound on its AEAD distinguishing advantage:

$$\mathsf{Adv}^{aead}_{\mathcal{O}_1}(\mathscr{A}) \leq \epsilon_{\mathsf{bad}} + \epsilon_{\mathsf{ratio}}.$$

A proof of this theorem is available in multiple papers including [11, 4, 9].

8.2 Some Results on Multicollision

8.2.1 Expected multicollition in an uniform sample

Let $\mathsf{X}_1, \ldots, \mathsf{X}_q \leftarrow_{\$} \mathcal{D}$ where $|\mathcal{D}| = N$. For notational simplicity, we write $\log_2 N$ as n. We denote the maximum multicollision random variable for the sample as $\mathsf{mc}_{q,N}$. More precisely, $\mathsf{mc}_{q,N} = \max_a |\{i : \mathsf{X}_i = a\}|$. For any integer $\rho \geq 2$,

$$\begin{aligned}
\Pr[\mathsf{mc}_{q,N} \geq \rho] &\leq \sum_{a \in \mathcal{D}} \Pr[|\{i : \mathsf{X}_i = a\}| \geq \rho] \\
&\leq N \cdot \frac{\binom{q}{\rho}}{N^\rho} \\
&\leq N \cdot \frac{q^\rho}{N^\rho \rho!} \\
&\leq N \cdot \left(\frac{qe}{\rho N}\right)^\rho
\end{aligned}$$

We justify the inequalities in the following way: The first inequality is due to the union bound. If there are at least ρ indices for which X_i takes value a, we can choose the first ρ indices in $\binom{q}{\rho}$ ways. This justifies the second inequality. The last inequality follows from the simple observation that $e^\rho \geq \rho^\rho/\rho!$. Thus, we have

$$\Pr[\mathsf{mc}_{q,N} \geq \rho] \leq N \cdot \left(\frac{qe}{\rho N}\right)^\rho. \tag{1}$$

For any positive integer valued random variable Y bounded by q:

$$\begin{aligned}
\mathsf{Ex}[\mathsf{Y}] &\leq \sum_{i=0}^{q} Pr[Y \geq i] \\
&\leq (\rho - 1) + \sum_{i=\rho}^{q} Pr[Y \geq i] \\
&\leq (\rho - 1) + \rho \sum_{j=1}^{\lceil \frac{q}{\rho} \rceil} Pr[Y \geq j \cdot \rho] \\
&\leq (\rho - 1) + \rho \sum_{j=1}^{\lceil \frac{q}{\rho} \rceil} N \cdot \left(\frac{qe}{j \cdot \rho N}\right)^{j \cdot \rho} \text{ substituting Eq (1)} \\
&\leq (\rho - 1) + \rho N \sum_{j=1}^{\lceil \frac{q}{\rho} \rceil} \left(\frac{qe}{\rho N}\right)^{j \cdot \rho}
\end{aligned}$$

Now if $\left(\frac{qe}{\rho N}\right) < 1$ then we have

$$\sum_{j=1}^{\lceil \frac{q}{\rho} \rceil} \left(\frac{qe}{\rho N}\right)^{j\cdot\rho} \leq \sum_{j=1}^{\infty} \left(\frac{qe}{\rho N}\right)^{j\cdot\rho} \leq \frac{\left(\frac{qe}{\rho N}\right)^{\rho}}{1-\left(\frac{qe}{\rho N}\right)^{\rho}}$$

Hence if $\left(\frac{qe}{\rho N}\right) < 1$

$$\mathsf{Ex}\left[\mathsf{Y}\right] \leq (\rho-1) + \rho N \cdot \frac{\left(\frac{qe}{\rho N}\right)^{\rho}}{1-\left(\frac{qe}{\rho N}\right)^{\rho}} \tag{2}$$

Using Eq. (1), and Eq. (2) we can prove the following results for the expected value of maximum multicollision. We write $\mathsf{mcoll}(q, N)$ to denote $\mathsf{Ex}\left[\mathsf{mc}_{q,N}\right]$.

Proposition 1. $\mathsf{mcoll}(q, N) < \begin{cases} \frac{4n}{\log n} & \textit{if } q = N, n \geq 16 \\ 4n & \textit{if } q = nN, n \geq 4 \\ 4n\lceil \frac{q}{nN} \rceil & \textit{if } q \geq nN, n \geq 4 \\ 4\log q & \textit{if } q < N, n \geq 16 \end{cases}$

Proof. First let $q = N$. Substituting q in Eq. 2 we have

$$\mathsf{Ex}\left[Y\right] \leq (\rho-1) + \rho N \cdot \frac{\left(\frac{e}{\rho}\right)^{\rho}}{1-\left(\frac{e}{\rho}\right)^{\rho}}$$

Now Let $\rho = \frac{4n}{logn}, n \geq 16$ Then $\frac{e}{\rho} < \frac{1}{2}$ and Hence $1 - \left(\frac{e}{\rho}\right)^{\rho} > \frac{e}{\rho}$. Hence

$$\mathsf{Ex}\left[Y\right] < (\rho-1) + \rho N \cdot \left(\frac{e}{\rho}\right)^{\rho-1}$$

Now by substituting the value of ρ in $\rho N \cdot \left(\frac{e}{\rho}\right)^{\rho-1}$ and by taking logarithm it can be easily shown that $\rho N \cdot \left(\frac{e}{\rho}\right)^{\rho-1} \leq 1$ and hence $\mathsf{Ex}\left[Y\right] < \frac{4n}{logn}, n \geq 16$.

Let $q = nN, \rho = 4n$. Substituting q, ρ in Eq. 2 we have

$$\mathsf{Ex}\left[Y\right] \leq (4n-1) + 4nN \cdot \frac{\left(\frac{e}{4}\right)^{4n}}{1-\left(\frac{e}{4}\right)^{4n}}$$

Now let $n \geq 4$ then we have $4n \leq N$ and hence

$$4nN \cdot \frac{\left(\frac{e}{4}\right)^{4n}}{1-\left(\frac{e}{4}\right)^{4n}} \leq N^2 \cdot \frac{\left(\frac{e}{4}\right)^{4n}}{1-\left(\frac{e}{4}\right)^{4n}}$$

Notice that for $n \geq 2$ we have

$$\frac{\left(\frac{e}{4}\right)^{4n}}{1-\left(\frac{e}{4}\right)^{4n}} < \left(\frac{e}{4}\right)^{\frac{18}{5}n} = \left[\left(\frac{e}{4}\right)^{\frac{18}{5}}\right]^{n} \leq \left(\frac{1}{4}\right)^{n} = \frac{1}{N^2}$$

The first ineqality follows from the facts that for $n \geq 2$

$$1-\left(\frac{e}{4}\right)^{4n} \geq 1-\left(\frac{e}{4}\right)^{8} > 9/10 \text{ and } \left(\frac{e}{4}\right)^{\frac{2}{5}n} \leq \left(\frac{e}{4}\right)^{\frac{4}{5}} < \frac{3}{4} \implies 1-\left(\frac{e}{4}\right)^{4n} > \left(\frac{e}{4}\right)^{\frac{2}{5}n}$$

Hence $\mathsf{Ex}\left[Y\right] < 4n, n \geq 4$.

When $q \geq nN$, we can group them into $\lceil q/nN \rceil$ samples each of size exactly nN (we can add more samples if required). This would prove the result when $q \geq nN$.

Finally, when $q < N$, we can simply bound $\mathsf{Ex}\left[\mathsf{mc}_{q,N}\right] < 4\log q$.

□

When $n \geq 16$, for all q, we can write the bounds into one single form:

$$\mathsf{mcoll}(q, N) < nq/N \tag{3}$$

8.2.2 Expected Maximum Multicollision in a Non-uniform Random Sample

Now we bound expectation of maximum multicollision in a sample $\mathsf{X}_1, \ldots, \mathsf{X}_q$ (can be arbitrarily dependent) which is not completely uniform random. However, it satisfies the following property for all distinct $i_1, \ldots, i_\rho$ for any integer $\rho \geq 2$:

$$\Pr(\mathsf{X}_{i_1} = a, \cdots \mathsf{X}_{i_\rho} = a) \leq \frac{1}{N'^r} \tag{4}$$

Then, we can actually perform the same analysis as before. For any integer $\rho \geq 2$, it can be shown that

$$\Pr[\mathsf{mc}_{q,N} \geq \rho] \leq N \cdot \left(\frac{qe}{\rho N'}\right)^\rho \tag{5}$$

Using it, we can prove the following results for expected value of maximum multicollision.

Proposition 2. $\mathsf{Ex}\left[\mathsf{mc}_{q,N}\right] < \begin{cases} 4\log q & \textit{if } q < N' \\ \frac{4n}{\log n} & \textit{if } N' \leq q < N'n \\ \frac{4q}{N'} & \textit{if } q \geq N'n \end{cases}$

In the non-random case, we denote $\mathsf{Ex}\left[\mathsf{mc}_{q,N}\right]$ by $\mathsf{mcoll}'(q, N)$ As before, when $n \geq 16$, we have

$$\mathsf{mcoll}'(q, N) \leq nq/N' \tag{6}$$

8.3 Multichain Security game

Let $\mathcal{L} = ((u_1, v_1), \ldots, (u_t, v_t))$ be a list of pairs of b-bit elements such that $\forall i \neq j, u_i \neq u_j, v_i \neq v_j$. For any such list we define $\mathsf{domain}(\mathcal{L}) = \{u_1, \ldots, u_t\}$ and $\mathsf{range}(\mathcal{L}) = \{v_1, \ldots, v_t\}$.

Given a list $\mathcal{L}$ we define a directed graph $\mathcal{G}_\mathcal{L}$ as follows: $\mathsf{range}(\mathcal{L})$ is the set of vertices of $\mathcal{G}_\mathcal{L}$. There are to types of edges:

Given any $i, j \in [t]$, there exist a directed edge $v_i \xrightarrow{x} v_j$ where $x = v_i \oplus u_j$.

Given any $i, j \in [t]$ there exist a directed edge $v_i \underset{x}{\rightarrow} v_j \iff u_j = (\lfloor x \rfloor_r \oplus \lfloor v_i \rfloor_r) \| (\lceil x \rceil_c \oplus \alpha^{\delta_x} \cdot \lceil v_i \rceil_c)$ similarly we can extend this definition to define a labled walk $\mathcal{W}$ from ω_0 to ω_k by

$$\mathcal{W} : \omega_0 \xrightarrow{x_1} \omega_1 \xrightarrow{x_2} \omega_2 \cdots \omega_{k-1} \xrightarrow{x_k} \omega_k$$

We simple denote this by $\omega_0 \xrightarrow{x} \omega_k$ where $x = (x_1, \ldots, x_k)$. k is the length of the walk. Similary by $w_0 \underset{y}{\xrightarrow{x}} w_{k+1}$ we denote the walk $\omega_0 \xrightarrow{x} \omega_k \underset{y}{\rightarrow} \omega_{k+1}$.

8.3.1 Multichain Structure

Definition 1. *Let $r, \tau \leq b$ be some parameters. We say that a set of labled walks $\{\mathcal{W}_1, \ldots, \mathcal{W}_p\}$ forms a multi-chain of a given lable $x = (x_1, x_2, \ldots, x_k)$ in the graph $\mathcal{G}_\mathcal{L}$ if $\forall 1 \leq i \leq p$ We have $\mathcal{W}_i : u_i \underset{x_k}{\xrightarrow{(x_1, \ldots, x_{k-1})}} v_i$ such that $\forall 1 \leq i, j \leq p$; $\lfloor u_i \rfloor_r = \lfloor u_j \rfloor_r$; $\lfloor v_i \rfloor_\tau = \lfloor v_j \rfloor_\tau$. We call it a multi-chain of length k.*

Let W_k denote the maximum size of a multi-chain of length k (of a given lable x) induced by $\mathcal{L}$.

Now consider an adversary $\mathscr{A}$ interacting at most t times with $f^\pm$. Let (x_i, dir_i) denote ith query where $x_i \in \{0, 1\}^b$ and dir_i is either $+$ or $-$ (representing forward or inverse query). If $dir_i = +$, it gets response y_i as $f(x_i)$, else the response y_i is set as $f^{-1}(x_i)$. After t many interactions, we define a list $\mathcal{L}$ of pairs $(u_i, v_i)_i$ where $(u_i, v_i) = (x_i, y_i)$ if $dir_i = +$, and $(u_i, v_i) = (y_i, x_i)$ otherwise. So we have $f(u_i) = v_i$ for all i. We call the tuple of triples $\theta := ((u_1, v_1, dir_1), \ldots, (u_t, v_t, dir_t))$ the transcript of the adversary $\mathscr{A}$ interacting with $f^\pm$. We also write $\theta' = ((u_1, v_1), \ldots, (u_t, v_t))$ which only stores the information about the random permutation. We write

$$\mu_{k,\mathscr{A}} := \mathsf{Ex}\left[W_k\right].$$

Here W_k is defined for the labeled graph induced by the list θ' as defined above and expectation is defined over the randomness of the random permutation f and the random coin of the adversary $\mathscr{A}$. Finally, we define $\mu_{k,t} = \max_\mathscr{A} \mu_{k,\mathscr{A}}$ where maximum is taken over all adversaries making at most t queries.

9 Security Proof of ORANGE-Zest

Recently, Dobraunig *et al.* came up with a practical forgery attack on ORANGE-Zest. Hence, we claim that the ORANGE-Zest at it's original version is not secure. The forgery attack exploits the fact that the initial input of the extra state, while processing the first message block, is always set to be K, and hence it is nonce independent. However, this attack can be avoided, simply by taking the initial extra state input in such a way that it depends on the nonce. In this regard, we propose to set the value as the most significant half of the permutation output received before processing the last associated data block. To ensure that this can be done in all the cases, when $|A| = 0$ and $|M| \neq 0$, the associated data is padded by $0^{n-1}1$ and is treated as a partial block. We note that if applying the above modification, we can take the same initial chain input $K\|N$ for all the cases where $|A| \neq 0$ or $|M| \neq 0$. .

For the sake of completeness, we provide a complete algorithm implementing the above modifications and then give a security proof for the modified design.

Algorithm 3 The Modified algorithm for ORANGE-Zest.

function ORANGE-ZEST$_{[\mathsf{P}]}$.enc(K, N, A, M)
  $(A_{a-1}, \ldots, A_0) \xleftarrow{n} A$
  $(M_{m-1}, \ldots, M_0) \xleftarrow{n} M$
  if $a = 0, m = 0$ **then**
    $(T, *) \leftarrow \mathsf{P}((K \oplus 2)\|N)$
    return (λ, T)
  if $a = 0, m \neq 0$ **then**
    $U \leftarrow P(K\|N)$
    $S \leftarrow \lceil U \rceil_n$
    $U \leftarrow \mathsf{mult}(2, U) \oplus 1$
    $(C, U) \leftarrow \mathsf{proc\_txt}(S, U, M, +)$
    return $(C, \mathsf{proc\_tg}(U))$
  $C \leftarrow \lambda$
  if $a \neq 0$ **then** $(U, S) \leftarrow \mathsf{proc\_hash}(K\|N, A, 1, 2)$
  if $a \neq 0, m \neq 0$ **then** $(C, U) \leftarrow \mathsf{proc\_txt}(S, U, M, +)$
  return $(C, \mathsf{proc\_tg}(U))$

function ORANGISH(D)
  $(D_{d-1}, \ldots, D_0) \xleftarrow{n} D$
  $D_d \leftarrow (n \nmid |D_{d-1}|)?\ 0^{n-2}10 : 0^{n-1}1$
  $D_{d-1} \leftarrow \mathsf{pad}(D_{d-1})$
  $X \leftarrow (0^n\|D_0)$
  for $i = 0$ to $d-1$ **do**
    $A_i \leftarrow (D_i\|D_{i+1})$
  $(Z, \star) \leftarrow \mathsf{proc\_hash}(X, (A_{d-1}\| \ldots \|, A_0), 0, 0)$
  $Z_1 \leftarrow \mathsf{P}(Z)$
  $Z_2 \leftarrow \mathsf{P}(Z_1)$
  return $\lfloor Z_2 \rfloor_n \| \lfloor Z_1 \rfloor_n$

function $\mathsf{proc\_txt}(S_0, U_0, D, \mathsf{dir})$
  $(D_{d-1}, \ldots, D_0) \xleftarrow{2n} D$
  for $i = 0$ **to** $d-1$ **do**
    $V_i \leftarrow \mathsf{P}(U_i)$
    if $i = d-1$ **then**
      $c \leftarrow (2n \mid |D_{d-1}|)?1 : 2$
      $V_i \leftarrow \mathsf{mult}(c, V_i)$
    $KS_i \leftarrow \rho(S_i, V_i)$
    $D'_i \leftarrow D_i \oplus \lfloor KS_i \rfloor_{|D_i|}$
    if $\mathsf{dir} = " + "$ **then** $D_i \leftarrow D'_i$
    $S_{i+1} \leftarrow \lceil V_i \rceil_n$
    $U_{i+1} \leftarrow V_i \oplus \mathsf{pad}(D_i)$
  return (D', U_d)

function ORANGE-ZEST$_{[\mathsf{P}]}$.dec(K, N, A, C, T)
  $(A_{a-1}, \ldots, A_0) \xleftarrow{n} A$
  $(C_{m-1}, \ldots, C_0) \xleftarrow{n} C,\ M \leftarrow \lambda$
  if $a = 0, m = 0$ **then** $(T', *) \leftarrow \mathsf{P}((K \oplus 2)\|N)$
  if $a = 0, m \neq 0$ **then**
    $U \leftarrow P(K\|N)$
    $S \leftarrow \lceil U \rceil_n$
    $U \leftarrow \mathsf{mult}(2, U) \oplus 1$
    $(M, U) \leftarrow \mathsf{proc\_txt}(S, U, C, -)$
    $T' \leftarrow \mathsf{proc\_tg}(U)$
  if $a \neq 0$ **then** $(U, S) \leftarrow \mathsf{proc\_hash}(K\|N, A, 1, 2)$
  if $a \neq 0, m \neq 0$ **then** $(M, U) \leftarrow \mathsf{proc\_txt}(S, U, C, -)$
  $T' \leftarrow \mathsf{proc\_tg}(U)$
  if $T \neq T'$ **then**
    return $\perp$
  else
    return $(M, \top)$

function $\mathsf{proc\_hash}(X, D, c_0, c_1)$
  $(D_{d-1}, \ldots, D_0) \xleftarrow{2n} D$
  $X_0 \leftarrow X$
  for $i = 0$ **to** $d-2$ **do**
    $Y_i \leftarrow \mathsf{P}(X_i)$
    $X_{i+1} \leftarrow Y_i \oplus D_i$
  $c \leftarrow (2n \mid |D_{d-1}|)?c_0 : c_1$
  $Y_{d-1} \leftarrow \mathsf{P}(X_{d-1})$
  $S \leftarrow \lceil Y_{d-1} \rceil_n$
  $Y_{d-1} \leftarrow \mathsf{mult}(c, Y_{d-1})$
  $X_d \leftarrow Y_{d-1} \oplus \mathsf{pad}(D_{d-1})$
  return (X_d, S)

function $\rho(S, Y)$
  $(Y^b, Y^t) \xleftarrow{n} Y$
  $Z \leftarrow (Y^b \oplus \alpha S)\|(Y^t \lll 1)$
  return Z

function $\mathsf{mult}(c, V)$
  $(V^b, V^t) \xleftarrow{n} V$
  return $\alpha^c \cdot V^b \| V^t$

function $\mathsf{proc\_tg}(U)$
  $(U^b, U^t) \xleftarrow{n} U$
  return $\mathsf{P}(U^t\|U^b)$

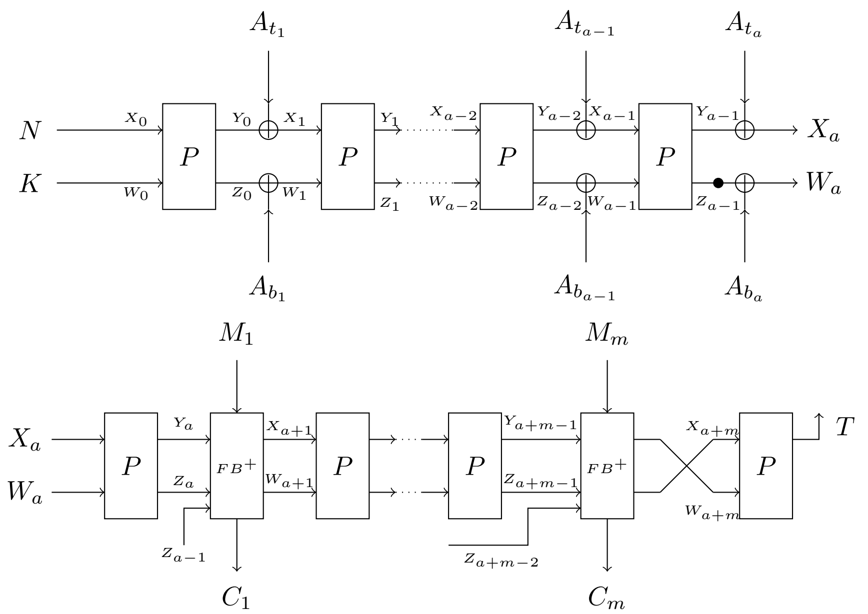

Figure 8: modified-ORANGE-Zest. Note that this is a representation of Figure 5 where the b-bit chains are separately shown as two $\frac{b}{2}$-bit chains with the only modification that the first extra state input is Z_{a-1} instead of K.

We fix a deterministic non-repeating query making distinguisher $\mathcal{A}$ that interacts with either (1) the real oracle $(\mathcal{O}^f, f)$ or (2) the ideal oracle $(\$^f, f)$ making at most,

1. q_e encryption queries $(N^i, A^i, M^i)_{i\in(q_e]}$ with an aggregate of total σ_e many blocks.
2. q_f offline or direct forward queries $(U^i, V^i, +)_{i\in(q_f]}$ to f.
3. q_b direct backward queries $(U^i, V^i, -)_{i\in(q_b]}$ to f.
4. attempts to forge with q_v many queries $(N^{\star i}, A^{\star i}, C^{\star i}, T^{\star i})_{i\in(q_v]}$ having a total of σ_v many blocks.

We assume $q_p = q_f + q_b$ to be the total number of offline or direct queries to f. Also for simplicity assume that, $\forall i, M^i$ and A^i have m_i and 0 many blocks respectively and $C^{\star i}$ and $A^{\star i}$ have m_i and 0 many blocks respectively. Let $X^\star, Y^\star, Z^\star, W^\star$ corresponds to the imtermidiate variables of the forging queries. Let $\mathcal{E}, \mathcal{D}$ denotes the sets of indices of the encryption and decryption queries.

Theorem 2. *For any $(q_p, q_e, q_v, \sigma_e, \sigma_v)-$ adversary $\mathscr{A}$ we have*

$$\mathsf{Adv}^{aead}_{\textit{ORANGE-Zest}}(\mathscr{A}) \le \frac{q_p}{2^\kappa} + \frac{5\sigma_e q_p}{2^b} + \frac{4\sigma_v q_p}{2^b} + \frac{2q_v}{2^\tau} + \frac{2rq_p\sigma_e}{2^b} + \frac{4\sigma_e\sigma_v}{2^c} + \sum_{i\in\mathcal{D}} \frac{\mu_{m_i,q_p}}{2^c} + \frac{rq_p\sigma_v\sigma_e}{2^{b+c}}$$

9.1 The Ideal World and Bad Transcript

In the ideal world there are three types of oracle queries, namely primitive query, encryption query and decryption query.

<u>Primitive Queries</u> The ideal world simulates $Q_\pm$ queries honestly and maintain a list ω_p of the query responce of Q as a partial injective list. More precisely

$$\omega_p = ((U^1, V^1, dir_1), (U^2, V^2, dir_2), \dots)$$

where $dir_i = +1$ for a direct forward query and -1 for a direct backward query. We keep ω_p as a list of direct forward queries. i.e. $f(U^i) = V^i$ for all i. Let $\omega_{p'} = (((U^1, V^1), (U^2, V^2), \dots)$ i.e. ω_p without considering the sign of the query.

<u>Encryption Queries</u> When the i-th query is an encryption query (N^i, M^i) where and $M^i = M^i_{m_i}\|\cdots\|M^i_2\|M^i_1$ it first defines

$$\delta^i_j = \begin{cases} 0 \text{ for } j < m_i \\ 1 \text{ for } j = m_i, |M^i_{m_i}| = b \\ 2 \text{ otherwise} \end{cases}$$

Then it samples $(Y^i_{-1}, Y^i_0, \ldots, Y^i_{m_i}) \xleftarrow{\$} \{0,1\}^r$ and $(Z^i_{-1}, Z^i_0, \ldots, Z^i_{m_i}) \xleftarrow{\$} \{0,1\}^c$, and returns $T = \lfloor Z^i_{m_i} \| Y^i_{m_i} \rfloor_\tau$ and $C^i = C^i_{m_i} \| \cdots \| C^i_2 \| C^i_1$ where for $1 \leq j \leq m$

$$C^i_{r_j} = M^i_{r_j} \oplus lRot(Y^i_{j-1});$$

$$C^i_{c_j} = M^i_{c_j} \oplus \alpha^{\delta^i_j} \cdot Z^i_{j-1} \oplus \alpha \cdot Z^i_{j-2}$$

$$C^i_j = C^i_{c_j} \| C^i_{r_j}$$

The intermidiate values X^i_j, W^i_j are calculated as follows:

$$X^i_j = \begin{cases} \lfloor K \| N \rfloor_r \text{ for } j = -1 \\ Y^i_{-1} \oplus 1 \text{ for } j = 0 \\ Y^i_{j-1} \oplus C^i_{r_j} \text{ for } 1 < j < m_i \\ \alpha^{\delta^i_{m_i}} Z^i_{m_i-1} \oplus C^i_{c_{m_i}} \text{ for } j = m_i \end{cases}$$

$$W^i_j = \begin{cases} \lceil K \| N \rceil_c \text{ for } j = -1 \\ \alpha^2 . Z^i_{-1} \text{ for } j = 0 \\ Z^i_{j-1} \oplus C^i_{c_j} \text{ for } j < m_i \\ Y^i_{m_i-1} \oplus C^i_{r_{m_i}} \text{ for } j = m_i \end{cases}$$

Decryption Queries When the i-th query is a decryption query of the form (N^{*i}, C^{*i}, T^{*i}) it always returns $M^{*i} = \perp$. The decryption transcript $\omega_d = (M^{*i})_{i \in \mathcal{D}}$ where $M^{*i} = \perp$ for all $i \in \mathcal{D}$

Offline Queries After all the above queries, finally the oracle returns all the X, Y, Z, W values defined above. Let $\omega_e := (X^i_j, Y^i_j, Z^i_j, W^i_j)_{i \in \mathcal{E}, j \in [m_i]}$. The transcript of the ideal oracle is $(\omega_p, \omega_e, \omega_d)$.

Intermediate Values of the decryption queries Given the i-th decryption query $(N^{*i}, C^{*i}, T^{*i}), i \in \mathcal{D}$ we define p_i as follows.

$$p_i = \begin{cases} -1 \text{ if } N^{*i} \neq N^{i'} \forall i' \in \mathcal{E} \\ l_i \text{ if } \exists i' \in \mathcal{E} \ni N^{*i} = N^{i'}; C^{*i}_j = C^{i'}_j \forall 1 \leq j \leq l_i < m_i; C^{*i}_{l_i+1} \neq C^{i'}_{l_i+1} \\ l_i - 1 \text{ otherwise} \end{cases}$$

Given a statement P let

$$\chi(P) = \begin{cases} 1 \text{ if } P \text{ is true} \\ 0 \text{ otherwise} \end{cases}$$

Let

$$x^i_j = \chi(j \neq m_i) Y^{*i}_{j-1} \oplus \chi(j = m_i) \alpha^{\delta^i_{m_i}} Z^{*i}_{j-1}.$$

$$w^i_j = \chi(j = m_i) Y^{*i}_{j-1} \oplus \chi(j \neq m_i) Z^{*i}_{j-1}.$$

For any $i \in \mathcal{D}$ we define, $\forall 0 \leq j \leq p_i$

$$X^{*i}_j = X^{i'}_j; Y^{*i}_j = Y^{i'}_j; Z^{*i}_j = Z^{i'}_j; W^{*i}_j = W^{i'}_j$$

Now we further extend X, Y, Z, W values using primitive transript wherever possible. For notational simplicity let $c^i_j := \chi(j \neq m_i) C^{*i}_j \oplus \chi(j = m_i) \lfloor C^{*i}_j \rfloor_r \| \lceil C^{*i}_j \rceil_c, \forall p_i < j \leq m_i$. If there exist a labeled walk in the labled directed graph induced by ω_p from $Z^{*i}_{p_i} \| Y^{*i}_{p_i}$ with lable $(c^i_{p_i+1}, \ldots, c^i_j), j < m_i$, then we denote the end node as $Z^{*i}_j \| Y^{*i}_j$.

$$Z^{*i}_{p_i} \| Y^{*i}_{p_i} \xrightarrow{(c^i_{p_i+1}, \ldots, c^i_j)} Z^{*i}_j \| Y^{*i}_j$$

given $i \in \mathcal{D}$ let $p'_i < m_i$ be the maximum possible value of such j.

For all such $i \in \mathcal{D}$ and $p_i < j \leq p'_i + 1$ define

$$X^{*i}_j = x^i_j \oplus \lfloor c^i_j \rfloor_r$$

$$W^{*i}_j = w^i_j \oplus \lceil c^i_j \rceil_c$$

9.2 Identifying bad events

We say that an ideal world transcript $\omega = (\omega_p, \omega_e, \omega_d)$ is bad if any one of the following conditions holds:

Bad events due to encryption and primitive transcript:

B1: For some $(U,V) \in \omega_p$, $K = \lceil U \rceil_\kappa$.

B2: For some $i \in \mathcal{E}$, $j \in [m_i]$, $Z^i_j \| Y^i_j \in \mathsf{range}(\omega_p)$, (in other words, $\mathsf{range}(\omega_e) \cap \mathsf{range}(\omega_p) \neq \emptyset$)

B3: For some $i \in \mathcal{E}$, $j \in [m_i]$, $W^i_j \| X^i_j \in \mathsf{domain}(\omega_p)$, (in other words, $\mathsf{domain}(\omega_e) \cap \mathsf{domain}(\omega_p) \neq \emptyset$)

B4: For some $(i \in \mathcal{E}, j \in [m_i]) \neq (i' \in \mathcal{E}, j' \in [m_{i'}])$, $Z^i_j \| Y^i_j = Z^{i'}_{j'} \| Y^{i'}_{j'}$,

B5: For some $(i \in \mathcal{E}, j \in [m_i]) \neq (i' \in \mathcal{E}, j' \in [m_{i'}])$, $W^i_j \| X^i_j = W^{i'}_{j'} \| X^{i'}_{j'}$,

Bad events due to decryption transcript:

B6: For some $i \in \mathcal{D} \ni p_i \leq m_i - 1$, $(i' \in \mathcal{E}, j' \in [m_{i'}])$, $W^{*i}_{p_i+1} \| X^{*i}_{p_i+1} = W^{i'}_{j'} \| X^{i'}_{j'}$,

B7: For some $i \in \mathcal{D}$ with $p_i \geq 0$, $p'_i = m_i - 1$ and $(W^{*i}_{m_i} \| X^{*i}_{m_i}, * \| T^{*i}) \in \omega_p$,

B8: For some $i \in \mathcal{D}$ with $p_i \geq 0$ and $p'_i \geq p_i + 1$, $W^{*i}_{p'_i+1} \| X^{*i}_{p'_i+1} \in \mathsf{domain}(\omega_e)$.

We write BAD to denote the event that the ideal world transcript Θ_0 is bad. Then, with a slight abuse of notations, we have

$$\mathsf{BAD} = \cup_{i=1}^{8} \mathtt{Bi}$$

Lemma 1.

$$\Pr[\mathsf{BAD}] \leq \frac{q_p}{2^\kappa} + \frac{5\sigma_e q_p}{2^b} + \frac{2rq_p\sigma_e}{2^b} + \frac{4\sigma_e\sigma_v}{2^c} + \sum_{i \in \mathcal{D}} \frac{\mu_{m_i,q_p}}{2^c} + \frac{rq_p\sigma_v\sigma_e}{2^{b+c}}$$

9.3 The Real World and Good Transcript Analysis

The real world has the oracle $f^{\pm}$. The AE encryption and decryption queries and direct primitive queries are faithfully responded based on $f\pm$. Like the ideal, after completion of interaction, the ideal oracle returns all Y, Z -values corresponding to the encryption queries only. Note that a decryption query may return M^i which is not $\perp$.

Now consider a good transcript $\omega = (\omega_p, \omega_e, \omega_d)$. The understanding of the bad events will becoe clear from understanding of the good transcript. Suppose for all $1 \leq j \leq p'_i$, Y^{*i}_j, Z^{*i}_j and $X^{*i}_{j+1}, W^{*i}_{j+1}$ have been defined as described above. Then observe the following:

1. The tuples ω_e is permutation compatible and disjoint from ω_p. So union of tuples $\omega_e \cup \omega_p$ is also permutation compatible.

2. For all $i \in \mathcal{D}$ we have either $p'_i = m_i - 1$ and $(W^{*i}_{m_i} \| X^{*i}_{m_i}, \star \| T^{*i}) \in \omega_p \cup \omega_e$ (Type-1 decryption query) or $p'_i < m_i - 1$ but $(W^{*i}_{p'_i+1} \| X^{*i}_{p'_i+1} \notin \omega_p \cup \omega_e$(Type-2 decryption query). Type-1 decryption queries would be e straightaway rejected. Type-2 decryption query can be computed based on $\omega_p \cup \omega_e$ until $(W^{*i}_{p'_i+1} \| X^{*i}_{p'_i+1}$ which is fresh. So $f(W^{*i}_{p'_i+1} \| X^{*i}_{p'_i+1})$ is random over a large set. This would ensure with high probability we reject those decryption queries also.

Based on the above observations we perform our analysis of the good transcripts.

<u>Good Transcript Analysis:</u> Now fix a good transcript ω. Let Θ_0 and Θ_1 denote the transcript random variable obtained in the ideal world and real world respectively. As noted before, all the input-output pairs for the underlying permutation are compatible. In the ideal world, all the Y, Z values are sampled uniform at random; the list ω_p is just the partial representation of f; and all the decryption queries are degenerately aborted; whence we get

$$\Pr[\Theta_0 = w] \leq \frac{1}{2^{b\sigma_e}(2^b)_{q_p}}$$

Here σ_e denotes the total number of blocks present in all encryption queries including nonce. In notation $\sigma_e = q_e + \sum_i m_i$.

In the real world, for ω we denote the encryption query, decryption query, and primitive query tuples by ω_e, ω_d and ω_p, respectively. Then, we have

$$\begin{aligned}
\Pr[\Theta_1 = \omega] &= \Pr[\Theta_1 = (\omega_e, \omega_p, \omega_d)] \\
&= \Pr[\omega_e, \omega_p] \cdot \Pr[\omega_d \mid \omega_e, \omega_p] \\
&= \Pr[\omega_e, \omega_p] \cdot (1 - \Pr[\neg\omega_d \mid \omega_e, \omega_p]) \\
&\le \Pr[\omega_e, \omega_p] \cdot \left(1 - \sum_{i \in \mathcal{D}} \Pr[\neg\omega_{d,i} \mid \omega_e, \omega_p]\right) \qquad (7)
\end{aligned}$$

Here we have slightly abused the notation to use $\neg\omega_{d,i}$ to denote the event that the i-th decryption query successfully decrypts and and $\neg\omega_d$ is the union $\cup_{i\in\mathcal{D}}\neg\omega_{d,i}$ (i.e. at least one decryption query successfully decrypts). The encryption and primitive queries are mutually permutation compatible, so we have

$$\Pr_{\Theta_1}(\omega_e, \omega_p) = 1/(2^b)_{\sigma_e+q_p} \ge \Pr_{\Theta_0}(\omega_e, \omega_p).$$

Now we show an upper bound $\Pr_{\Theta_1}(\neg\omega_{d,i} \mid \omega_e, \omega_p) \le \frac{m_i(\sigma_e+q_p)}{2^b-\sigma_e-q_p} + \frac{1}{2^\tau}$ for every type-2 decryption query. Recall that $W^{*i}_{p'_i+1} \| X^{*i}_{p'_i+1}$ is fresh. If $W^{*i}_j \| X^{*i}_j$ is the last input block then $f(W^{*i}_j \| X^{*i}_j) = * \| T^{*i}$ with probability at most $2/2^\tau$ (provided $\sigma_e + q_p \le 2^{b-1}$ which can be assumed, since otherwise our bound is trivially true). Suppose $W^{*i}_j \| X^{*i}_j$ is not the last block, then the next input block may collide with some encryption or primitive input block with probability at most $\frac{\sigma_e+q_p}{2^b}$. Applying this same argument for all the successive blocks till the last one, we get the probability at most $\frac{m_i(\sigma_e+q_p)}{2^b-\sigma_e-q_p}$, the last block input would be fresh. Hence the probability that the tag matches is at most $2/2^\tau$. Now, by union bound we have

$$\begin{aligned}
\Pr[\neg\omega_d \mid \omega_e, \omega_p] &\le \sum_{i\in\mathcal{D}} \frac{m_i(\sigma_e + q_p)}{2^b - \sigma_e - q_p} + \frac{2}{2^\tau} \\
&\le \frac{2\sigma_v(\sigma_e + q_p)}{2^b} + \frac{2q_v}{2^\tau} \\
&\le \frac{4\sigma_v q_p}{2^b} + \frac{2q_v}{2^\tau}.
\end{aligned}$$

We have Theorem 2. follows from Equation 7, Lemma 1 and Theorem 1. □

9.4 Bounding bad events(Proof of Lemma 1)

bounding $\Pr[\mathtt{B1}]$: Fix $i \in (q_p]$.Since K is randomly chosen, probability of $(U^i, V^i) \in \omega_p$ s.t. $\lfloor U^i \rfloor_\kappa = K$ is bounded by $\frac{1}{2^\kappa}$. Hence bounding over all i, we have

$$\Pr[\mathtt{B1}]] \le \frac{q_p}{2^\kappa}$$

.

bounding $\Pr[\mathtt{B2}]$: This event can be analysed by deviding in the following cases

Case 1. $\exists i, j, a; Z^i_j \| Y^i_j = V_a$. Encryption after primitive query : This case can be bounded by probability at most $\frac{1}{2^b}$. Running over q_p many primitive queries and σ_e many blocks we have

$$\Pr[Case1] \le \frac{q_p \cdot \sigma_e}{2^b}$$

Case 2. $\exists i, j, a; Z^i_j \| Y^i_j = V_a, dir_a = +$. Encryption before primitive query This can be bounded by probability atmost $\frac{1}{2^b-a+1}$ Running over σ_e many encryption blocks and q_f many a indices we have

$$\Pr[Case2] \le \frac{q_f \cdot \sigma_e}{2^b - a + 1}$$

Case 3. $\exists i, j, a; Z^i_j \| Y^i_j = V_a, dir_a = -$. Encryption before primitive query Here the adversary has access to Y^i_j as it has already been released. Let Φ_{out} denote the number of multicollision in Y^i_j.

$$
\begin{aligned}
\Pr\left[Case3\right] &= \sum_{\Phi_{out}} \Pr\left[Case3 \wedge \Phi_{out}\right] \\
&= \sum_{\Phi_{out}} \Pr\left[Case3|\Phi_{out}\right] \cdot \Pr\left[\Phi_{out}\right] \\
&\leq \sum_{\Phi_{out}} \frac{\Phi_{out}.q_b}{2^c} \Pr\left[\Phi_{out}\right] \\
&\leq \frac{q_p}{2^c} \sum_{\Phi_{out}} \Phi_{out} \Pr\left[\Phi_{out}\right] \\
&\leq Ex[\Phi_{out}] \cdot \frac{q_p}{2^c} = \frac{q_p \cdot \mathsf{mcoll}(\sigma_e, 2^r)}{2^c}
\end{aligned}
$$

Since the three Cases are mutually exclusive, we have,

$$\Pr\left[\mathtt{B2}\right] \leq \frac{2 \cdot q_p \cdot \sigma_e}{2^b} + \frac{q_p \cdot \mathsf{mcoll}(\sigma_e, 2^r)}{2^c}$$

bounding $\Pr\left[\mathtt{B3}\neg\mathtt{B1}\right]$: Case 1: $\exists i, j, a,\ W^i_j \| X^i_j = U_a$, encryption after primitive: This case can be bounded by probability at most $1/2^b$, as Y^i_{j-1} and Z^i_{j-1} are chosen uniformly at random and hence X^i_j and W^i_j are determined randomly . We have at most σ_e many (i, j) pairs and q_p many a indices. Thus this can be bounded by at most $\sigma_e q_p/2^b$.
Case 2: $\exists i, j, a,\ W^i_j \| X^i_j = U_a,\ dir_a = -$, encryption before primitive: This case can be bounded by probability at most $1/(2^b - a + 1)$. We have at most σ_e many (i, j) pairs and q_b many a indices. Thus this can be bounded by at most $\sigma_e q_b/(2^b - a + 1)$.
Case 3: $\exists i, j, a,\ W^i_j \| X^i_j = U_a,\ dir_a = +$, encryption before primitive: Let Φ_{in} denote the number of multicollisions on X^i_j.

With a similar analysis on the multicollision of output values, we have $Pr[\text{Case 3}] \leq \mathsf{Ex}\left[\Phi\right]_{in} \frac{q_b}{2^c}$. Since the three cases are mutually exclusive, we have

$$Pr[\mathtt{B3}\neg\mathtt{B1}] \leq \frac{2\sigma_e q_p}{2^b} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c}.$$

Bounding $\Pr[\mathtt{B4}]$: The probability of this event can be bounded in a straightforward manner by at most $\sigma_e(\sigma_e - 1)/2^{b+1}$.
Bounding $\Pr[\mathtt{B5}]$: This event is similar to $\mathtt{B4}$, and the probability is bounded by at most $\sigma_e(\sigma_e - 1)/2^{b+1}$.
Bounding $\Pr[\mathtt{B6}]$: Note that after the i-th online query the adversary knows the following values;
$Y^i_{j-1}, X^i_j, Z^i_{j-1} \oplus \alpha Z^i_{j-2} = Z^i_{j-1} \oplus \alpha^j Z^i_{-1}\ \forall 1 \leq j \leq m_i - 1; Y^i_{m_i-1}, W^i_{m_i-1}, \alpha^{\delta^i_{m_i}} Z_{m_i-1} \oplus \alpha Z_{m_i-2} = \alpha^{\delta^i_{m_i}} Z_{m_i-1} \oplus \alpha^{m_i} Z^i_{-1}, T$.
Case 1. $p_i = m_i - 1, j' = m_{i'}; W^i_{m_i} \| X^i_{m_i} = W^{i'}_{m_{i'}} \| X^{i'}_{m_{i'}}$: The values of $W^i_{m_i} \| X^i_{m_i}$ and $W^{i'}_{m_{i'}} \| X^{i'}_{m_{i'}}$ upto r-most significant bits can be matched by adjusting $\lfloor C^{*i}_{p_i+1} \rfloor_r = \lfloor C^{i'}_{m_{i'}} \rfloor_r \oplus Y^{*i}_{m_i-1} \oplus Y^{i'}_{m_{i'}-1}$

Now We have $\lfloor W^i_{m_i} \| X^i_{m_i} \rfloor_c = \alpha^{\delta^i_{m_i}} Z^i_{m_i-1} \oplus \lceil C^{*i}_{m_i} \rceil_c$ and $\lfloor W^{i'}_{m_{i'}} \| X^{i'}_{m_{i'}} \rfloor_c = \alpha^{\delta^{i'}_{m_{i'}}} Z^{i'}_{m_{i'}-1} \oplus \lceil C^{i'}_{m_{i'}} \rceil_c$
Hence Case 1 happens iff

$$\lceil C^{i'}_{m_{i'}} \rceil_c = \alpha^{\delta^{i'}_{m_{i'}}} Z^{i'}_{m_{i'}-1} \oplus \alpha^{\delta^i_{m_i}} Z^i_{m_i-1} \oplus \lceil C^{*i}_{m_i} \rceil_c == \alpha^{m_i} Z^i_{-1} \oplus \alpha^{m_{i'}} Z^{i'}_{-1} \oplus \lceil C^{*i}_{m_i} \rceil_c \oplus A$$

Where A is some known value. Now if $N^{*i} \neq N^{i'}$ we have $Z^{i'}_{-1}, Z^i_{-1}$ are chosen indenendently at uniformly random, hence, we have probability that the above holds is atmost $\frac{1}{2^c}$. If $N^{*i} = N^{i'}$ then we must have $m_i \neq m_{i'}$ and hence since Z^i_{-1} is chosen at uniformly random, we have $\alpha^{m_i} Z^i_{-1} \oplus \alpha^{m_{i'}} Z^i_{-1}$ is uniformly random. Hence the probability is again atmost $\frac{1}{2^c}$. Varying over all $i \in \mathcal{D}$ and $i' \in \mathcal{E}$ we have

$$\Pr\left[\text{Case 1}\right] \leq \frac{q_v q_e}{2^c}$$

Case 2. $p_i = m_i - 1, j' < m_{i'}; W^i_{m_i} \| X^i_{m_i} = W^{i'}_{j'} \| X^{i'}_{j'}$:
We have $W^i_{m_i} \| X^i_{m_i} = (\lfloor C^{*i}_{p_i+1} \rfloor_r \oplus Y^{*i}_{m_i-1}) \| (\alpha^{m_i} Z^i_{-1} \oplus \lceil C^{*i}_{m_i} \rceil_c \oplus A)$
$W^{i'}_{j'} \| X^{i'}_{j'} = (\alpha^{j'} Z^{i'}_{-1} \oplus \lceil C^{i'}_{j'} \rceil_c \oplus B) \| (\lfloor C^{i'}_{j'} \rfloor_r \oplus Y^{i'}_{j'-1})$. Where A and B are known values.

If $r = c = \frac{b}{2}$ it can be seen that Case 2 holds iff $(\alpha^{m_i} Z^i_{-1} \oplus \lceil C^{*i}_{m_i} \rceil_c) = (\lfloor C^{i'}_{j'} \rfloor_r \oplus Y^{i'}_{j'-1})$ and $(\lfloor C^{*i}_{p_i+1} \rfloor_r \oplus Y^{*i}_{m_i-1}) = (\alpha^{j'} Z^{i'}_{-1} \oplus \lceil C^{i'}_{j'} \rceil_c)$ both holds. If $N^{*i} \neq N^{i'}$ we have $Z^{i'}_{j'-1}, Z^i_{m_i-1}$ are chosen independently uniformly at random, we have for fix i, i', j' the probablity is bounded by $\frac{1}{2^{2c}}$.

If $N^{*i} = N^{i'}$, We have since Z^i_{-1} is chosen uniformly at random and since both the equations need to hold independently we have again the probability is bounded by $\frac{1}{2^{2c}}$.

Now varying over all $i \in \mathcal{D}, i' \in \mathcal{E}, j' \in (m'_i]$ we have

$$\Pr[\text{case } 2] \leq \frac{q_v \sigma_e}{2^{2c}}$$

<u>Case 3. $p_i < m_i - 1, j' = m_{i'}; W^i_{m_i} \| X^i_{m_i} = W^{i'}_{j'} \| X^{i'}_{j'}$</u> : This can be bounded in the same way as in Case 2. by

$$\Pr[\text{case } 3] \leq \frac{q_v q_e}{2^{2c}}$$

<u>Case 4. $p_i < m_i - 1, j' < m_{i'}; W^i_{p_i+1} \| X^i_{p_i+1} = W^{i'}_{j'} \| X^{i'}_{j'}$:</u>

$X^i_{m_i}$ and $X^{i'}_{j'}$ can be matched by adjusting $\lfloor C^{*i}_{p_i+1} \rfloor_r = \lfloor C^{i'}_{j'} \rfloor_r \oplus Y^{*i}_{p_i} \oplus Y^{i'}_{j'-1}$

Now $W^i_{m_i}$ and $W^{i'}_{j'}$ matches iff

$$\lceil C^{i'}_{j'} \rceil_c = Z^i_{p_i} \oplus Z^{i'}_{j'-1} \oplus \lceil C^{*i}_{p_i+1} \rceil_c = \alpha^{p_i+1} Z^i_{-1} \oplus \alpha^{j'} Z^{i'}_{-1} \oplus \lceil C^{*i}_{p_i+1} \rceil_c \oplus A$$

Where A is some known value.

Now We have if $N^{*i} \neq N^{i'}$ then Z^i_{-1} and $Z^{i'}_{-1}$ are independent and chosen uniformly at random. If $N^{*i} = N^{i'}$ then we must have $p_i + 1 \neq j'$ and hence $\alpha^{p_i+1} Z^i_{-1} \oplus \alpha^{j'} Z^i_{-1}$ is uniformly random.

Hence, the probability that the above happen in the i-th query can be bounded by $\frac{\sigma_e}{2^c}$ and hence,

$$\Pr[\text{ Case } 4] \leq \frac{q_v \sigma_e}{2^c}$$

Since all the above cases are mutually exclusive we have

$$\Pr[\mathsf{B6}] \leq \frac{4\sigma_e \sigma_v}{2^c}$$

Bounding $\Pr[\mathsf{B7}]$: Let $W_k(\omega_{p'})$ denote the k-length multichain induced by ω_p. Suppose the event holds for the i-th decryption query and $N^{*i} = N^{i'}$. So $Z^{i'}_{p_i} \| Y^{i'}_{p_i}$ must be the starting node of the multi-chain. Since $Z^{i'}_{p_i}$ can be chosen randomly and independent of ω_p we have the probability to hold $\mathsf{B7}$ in the i-th decryption query is atmost $\frac{W_{m_i}}{2^c}$. So by union bound $\Pr[\mathsf{B7} | \omega_p] \leq \sum_{i \in \mathcal{D}} \frac{W_{m_i}}{2^c}$. Hence

$$\Pr[\mathsf{B7}] \leq \sum_{i \in \mathcal{D}} \frac{\mu_{m_i, q_p}}{2^c}$$

Bounding $\Pr[\mathsf{B8}]$: This event corresponds to the case when the first non-trivial decryption query block matches a primitive query and after following some partial chain matches an encryption query block. The probability of this event happening in the i-th decryption query is at most $\frac{q_p}{2^c} \times \frac{m_i^* \Phi_{in}}{2^c}$. Taking expectation we obtain

$$\Pr[\mathsf{B8}] \leq \frac{q_p \sigma_v \mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}$$

Lemma 1 can be proved by adding all the probabilities and bounding $\mathsf{mcoll}(\sigma_e, 2^r)$ by $\frac{r\sigma_e}{2^r}, \forall r \geq 16$. $\square$

9.5 Bounding Multichain

Theorem 3. *We have,*

$$\mu_{k,t} \leq \mathsf{mcoll}(t, 2^\tau) + \mathsf{mcoll}(t, 2^r) + k \cdot \mathsf{mcoll}'(t^2, 2^b)$$

Observation We have if $v_i \xrightarrow{x} v_j$ and $v_i \xrightarrow{x} v_k$ then $v_j = v_k$. Similarly if $v_i \xrightarrow[x]{} v_j$ and $v_i \xrightarrow[x]{} v_k$ then $v_j = v_k$. and hence if $v_i \xrightarrow[y]{x} v_j$ and $v_i \xrightarrow[y]{x} v_k$ then $v_j = v_k$.

<u>More Notations:</u> Let $W^{fwd,a}$ denote the size of the set $\{i : dir_i = +, \lfloor v_i \rfloor_\tau = a\}$ and $W^{fwd} = \max_a W^{fwd,a}$. This denotes the maximum number of multicollision in the τ- least significant bits of forward query responses.

Similarly define $W^{bck,a} = |\{i : dir_i = -, \lfloor u_i \rfloor_r = a\}|$ and $W^{bck} = \max_a W^{bck,a}$. This denotes the maximum number of multicollisions in the r- least significant bits of backward query responses.

Now Let $W^{mitm,a} = |\{(i,j) : v_i \xrightarrow{a} v_j \text{ or } v_i \xrightarrow[a]{} v_j\}|$ and $W^{mitm} = \max_a W^{mitm,a}$.

Lemma 2.

$$W_k \leq W^{fwd} + w^{bck} + k.W^{mitm}$$

Proof. Let $p = W_k$ and $\{\mathcal{W}_1, \ldots, \mathcal{W}_p\}$ be k-chains such that:

$$\forall 1 \leq i \leq p \mathcal{W}_i : v_0^i \xrightarrow[x_k]{(x_1,\ldots,x_{k-1})} v_k^i \text{ and}$$

$$\forall 1 \leq i \leq p; \lfloor v_0^i \rfloor_r = u; \lfloor v_k^i \rfloor_\tau = v.$$

Define

$$\omega_p^0 = |\{\mathcal{W}_i \in \{\mathcal{W}_1, \ldots, \mathcal{W}_p\} \mid (u_0^i, v_0^i, -) \in \theta\}|$$

$$\omega_p^{k+1} = |\{\mathcal{W}_i \in \{\mathcal{W}_1, \ldots, \mathcal{W}_p\} \mid (u_k^i, v_k^i, +) \in \theta\}|$$

$$\omega_p^j = |\{\mathcal{W}_i \in \{\mathcal{W}_1, \ldots, \mathcal{W}_p\} \mid (u_{j-1}^i, v_{j-1}^i, +) \in \theta \text{ and } (u_j^i, v_j^i, -) \in \theta\}| \; \forall 1 \leq j << k$$

Then clearly By union bound ;

$$W_k \leq \omega_p^0 + \omega_p^{k+1} + \sum_{j=1}^{k} \omega_p^j$$

Now by defininition of $W^{fwd}, W^{bck}, W^{mitm}$ we have,

$$\omega_p^0 \leq W^{fwd}; \omega_p^{k+1} \leq W^{bck}, \omega_p^j \leq W^{mitm}, \forall 1 \leq j \leq k.$$

□

Proof. (Theorem 3)

$$\mathsf{Ex}\left[W^{bck}\right] = \mathsf{Ex}\left[\mathsf{mc}_{t,2^r}\right] \leq \mathsf{mcoll}(t, 2^r) \leq \frac{rt}{2^r}$$

$$\mathsf{Ex}\left[W^{fwd}\right] = \mathsf{Ex}\left[\mathsf{mc}_{t,2^\tau}\right] \leq \mathsf{mcoll}(t, 2^\tau) \leq \frac{\tau t}{2^\tau}$$

$$\mathsf{Ex}\left[W^{mitm}\right] = \mathsf{Ex}\left[\mathsf{mc}_{t^2,2^b}\right] \leq \mathsf{mcoll}'(t^2, 2^b) \leq \frac{bt^2}{2^b}.$$

□

Combining Theorem 3 and Theorem 2 we get ,

Theorem 4. *(Main Result)*

$$\mathsf{Adv}^{aead}_{\mathit{ORANGE\text{-}Zest}}(\mathscr{A}) \leq \frac{q_p}{2^\kappa} + \frac{5\sigma_e q_p}{2^b} + \frac{4\sigma_v q_p}{2^b} + \frac{2q_v}{2^\tau} + \frac{2rq_p\sigma_e}{2^b} + \frac{4\sigma_e\sigma_v}{2^c} + \frac{rq_p\sigma_v\sigma_e}{2^{b+c}} + \frac{\tau q_v q_p}{2^{\tau+c}} + \frac{rq_v q_p}{2^b} + \frac{b\sigma_v q_p^2}{2^{b+c}}.$$

References

[1] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. In *ECRYPT hash workshop*, volume 2007. Citeseer, 2007.

[2] Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. On the indifferentiability of the sponge construction. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 181–197. Springer, 2008.

[3] Avik Chakraborti, Nilanjan Datta, Mridul Nandi, and Kan Yasuda. Beetle family of lightweight and secure authenticated encryption ciphers. *IACR Cryptology ePrint Archive*, 2018:805, 2018.

[4] Shan Chen and John Steinberger. Tight security bounds for key-alternating ciphers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 327–350. Springer, 2014.

[5] Tingting Cui, Ling Sun, Huaifeng Chen, and Meiqin Wang. Statistical integral distinguisher with multi-structure and its application on AES. In *Information Security and Privacy - 22nd Australasian Conference, ACISP 2017, Auckland, New Zealand, July 3-5, 2017, Proceedings, Part I*, pages 402–420, 2017.

[6] Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON family of lightweight hash functions. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 222–239. Springer, 2011.

[7] Jérémy Jean, María Naya-Plasencia, and Thomas Peyrin. Improved rebound attack on the finalist grøstl. In Anne Canteaut, editor, *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, volume 7549 of *LNCS*, pages 110–126. Springer, 2012.

[8] Jérémy Jean, María Naya-Plasencia, and Thomas Peyrin. Multiple limited-birthday distinguishers and applications. In *Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*, pages 533–550, 2013.

[9] Bart Mennink and Samuel Neves. Encrypted davies-meyer and its dual: Towards optimal security using mirror theory. In *Annual International Cryptology Conference*, pages 556–583. Springer, 2017.

[10] Yusuke Naito and Kazuo Ohta. Improved indifferentiable security analysis of PHOTON. In *Security and Cryptography for Networks - 9th International Conference, SCN 2014, Amalfi, Italy, September 3-5, 2014. Proceedings*, pages 340–357, 2014.

[11] Jacques Patarin. *Etude des generateurs de permutations pseudo-aleatoires bases sur le schema du d. E. S.* PhD thesis, Paris 6, 1991.

[12] Qingju Wang, Lorenzo Grassi, and Christian Rechberger. Zero-sum partitions of PHOTON permutations. In *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings*, pages 279–299, 2018.

[13] Hongjun Wu. The hash function jh. *Submission to NIST (round 3)*, 6, 2011.

Appendix A

Test vectors for ORANGE-Zest

Test vector 1:
$Key = 000102030405060708090A0B0C0D0E0F$
$Nonce = 000102030405060708090A0B0C0D0E0F$
$PT =$
$AD = 00010203$
$CT = 84A4C553119EA342C50CCCCE43782567$

Test vector 2:
$Key = 000102030405060708090A0B0C0D0E0F$
$Nonce = 000102030405060708090A0B0C0D0E0F$
$PT =$
$AD =$
$CT = 5A65624E01D1349D2211EFBD52217976$

Test vector 3:
$Key = 000102030405060708090A0B0C0D0E0F$
$Nonce = 000102030405060708090A0B0C0D0E0F$
$PT = 000102030405060708090A0B0C0D0E0F101112131415161718191A$
$AD = 000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D$
$CT = 06C8617CFB5C8CACA64F1F2B9460EADE7776AB0F814F4CFB0E561C621A$
$B9EB080D6CE0D200E80EE74E8C00$

Test vectors for **ORANGISH**

Test vector 1:
$Msg = 00010203$
$MD = 51390073EFBB1DEF2CEAD9688CC2C9D907F2EF6AC8C8D7E733$
$17EB2C28155226$

Test vector 2:
$Msg = 000102030405060708090A0B0C0D0E0F101112131415161718$
$191A1B1C1D1E1F202122232425262728292A2B2C2D2E2F3031323$
$33435363738393A3B3C3D3E3F404142434445464748494A4B4C4D4E4$
$F505152535455565758595A5B5C5D5E5F606162$
$MD = 7B1A8606FF708377BB612E0712C7E824921A8D78B9AD3258$
$A7B400E96AA349C3$

Test vector 3:
$Msg = 000102030405060708090A0B0C0D0E0F10111213141516171819$
$1A1B1C1D1E1F202122232425262728292A2B2C2D2E2F30313233343$
$5363738393A3B3C3D3E3F$
$MD = 85739793F2A59EC254488C3931447E86E0F3C0C919899DDA1B$
$F34B1639DFDCD8$