Improved Security for OCB3

Ritam Bhaumik$^{(\boxtimes)}$ and Mridul Nandi

Indian Statistical Institute, Kolkata, India
bhaumik.ritam@gmail.com, mridul.nandi@gmail.com

Abstract. OCB3 is the current version of the OCB authenticated encryption mode which is selected for the third round in CAESAR. So far the integrity analysis has been limited to an adversary making a single forging attempt. A simple extension for the best known bound establishes integrity security as long as the total number of query blocks (including encryptions and forging attempts) does not exceed the birthday-bound. In this paper we show an improved bound for integrity of OCB3 in terms of the number of blocks in the forging attempt. In particular we show that when the number of encryption query blocks is not more than birthday-bound (an assumption without which the privacy guarantee of OCB3 disappears), even an adversary making forging attempts with the number of blocks in the order of $2^n/\ell_{\mathrm{MAX}}$ (n being the block-size and ℓ_{MAX} being the length of the longest block) may fail to break the integrity of OCB3.

Keywords: OCB · OCB3 · Authenticated encryption · Integrity · Multiple verification query

1 Introduction

Authenticated encryption schemes [Rog02], which target both data confidentiality and integrity simultaneously, have received considerable attention in recent years. The increased interest is in part due to the ongoing CAESAR competition [cae], which aims to deliver a portfolio of state-of-the-art authenticated encryption schemes covering a spectrum of security and efficiency trade-offs. While other possibilities exist, it is natural to build AE schemes from blockciphers, employing some mode of operation. Some of the known blockcipher based authenticated encryptions are OCB [RBB03,Rog04], GCM [MV04,MV05], COPA [ABL+13], ELmD [DN14] and AEZ [HKR15]. Due to the CAESAR competition, many designs have appeared in literature. Moreover, some designs have been refined for better performance and improved security. OCB3 (submitted to CAESAR and now a third round candidate) is one such example, an enhancement of the well known construction OCB.

OCB and OCB3. OCB is a blockcipher-based mode of operation that achieves authenticated encryption in almost the same amount of time as the fastest conventional mode, CTR mode [WHF02], which achieves privacy alone in that time.

T. Takagi and T. Peyrin (Eds.): ASIACRYPT 2017, Part II, LNCS 10625, pp. 638–666, 2017.
https://doi.org/10.1007/978-3-319-70697-9_22

Despite this, OCB is simple and clean, and easy to implement in either hardware or software. For every message, it uses two blockcipher calls to process a constant block 0^n and a nonce, one blockcipher call for each message block and one additional blockcipher call for the checksum.

The refined OCB or $\mathsf{OCB3}$ [KR11] aims to shave off one AES encipherment per message encrypted about 98% of the time. The nonce here is used as a counter, i.e., in a given session, its top segment (of 122 bits) stays fixed, while, with each successive message, the bottom segment (of 6 bits) gets bumped up by one. This is the approach recommended in RFC 5116 [McG08, Sect. 3.2].

Known Security Results of $\mathsf{OCB3}$. Though the original OCB has already been proved to be secure, the security bound provided by [KR11], in particular the authenticity bound, does not show the standard birthday-bound security when the adversary is allowed to make multiple verification queries. More formally, the original bound is $O(\sigma_T^2/N) + O(1/2^\tau)$ where σ_T denotes the total number of input blocks in q encryption queries and τ denotes the tag length, if the number of verification queries is one. This bound generally implies $O(q'\sigma_T^2/N) + O(q'/2^\tau)$ when the number of verification queries is $q' \geq 1$, hence the provable security is degraded.

Another adaptation of the proof can be applied to obtain a bound of the form $O((\sigma_T + \sigma'_T)^2/N + q'/2^\tau)$, where σ'_T is the blocks in the decryption queries. It still remains birthday-bound in σ'_T. All known attacks exploit the collision in the input blocks for all encryption queries and hence σ_T^2/N is tight. But no matching attack with advantage $\sigma_T'^2/N$ is known. A recent collision attack on PHash [GPR17] (used to process the associated data) can be applied to obtain an integrity attack with advantage $O(\sigma'_T/N)$. (See [BGM04].)

Our Contribution. We show that the existing attack is the best possible by improving the integrity advantage. We follow the combined AE security distinguishing game [RS06] to bound the integrity security of OCB3. We use Patarin's coefficients H technique [Pat08] to bound the AE distinguishing game.

Theorem 1 (Main Result). *Let $\mathcal{A}$ be an adversary that makes q encryption queries consisting of σ message blocks in all with at most ℓ_{MAX} blocks per query, and α associated data blocks in all, and q' decryption queries in a nonce-respecting authenticated encryption security game with associated data against a real oracle $\mathcal{O}_1$ representing $\mathsf{OCB3}$ and an ideal oracle $\mathcal{O}_0$ representing an ideal nonce-based authenticated encryption function. Then*

$$\mathbf{Adv}^{\mathsf{naead}}_{\mathcal{O}_1,\mathcal{O}_0}(\mathcal{A}) \leq \frac{5\sigma_T^2}{N} + \frac{2\sigma^4}{N^2} + \frac{64q'\ell_{MAX} + 15q'}{N},$$

where $\sigma_T = \sigma + \alpha + q$ is the total number of blocks queried in the encryption queries (including messages, associated data and nonces).

2 Preliminaries

$\mathbb{N}$ denotes the set of non-negative integers. For $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, \ldots, n\}$ ($[0]$ is thus the empty set). For $m, n \in \mathbb{N}$, $[m..n]$ denotes the set $\{m, \ldots, n\}$ (which is the empty set when $m > n$). For a binary string $x \in \{0,1\}^*$, $|x|$ will denote the number of bits in x. We fix an arbitrary block-length $n \in \mathbb{N} \setminus \{0\}$. If $|x| = n$, we call x a complete block; if $|x| < n$, we call x an incomplete block; if $|x| = 0$ (the null string), we call x the empty block. (By convention, the empty block is also an incomplete block.) $\oplus$ and $\cdot$ denote the field addition (XOR) and field multiplication respectively over the finite field $\{0,1\}^n$. During calculations, for two block x and y, we will simply write $x + y$ to denote $x \oplus y$.

For $i \in [|x|]$, $x[i]$ denotes the i-th bit of x (we begin all indexing from 1, so $x[1]$ is the first bit of x). For $i \geq 1$ and $j \leq |x|$, $x[i..j]$ denotes the $(j - i + 1)$-bit contiguous substring of x starting at the i-th bit when $i \leq j$, and the empty string otherwise. For two strings x and y, $x||y$ denotes the concatenation of x and y. For a bit b, b^m denotes an m-bit string with each bit equal to b.

Any $x \in \{0,1\}^*$ can be mapped uniquely to a sequence $(x_1, \ldots, x_\ell, x_*)$, where $\ell \in \mathbb{N}$, $x_1, \ldots, x_\ell$ are complete blocks, and x_* is an incomplete (possibly empty) block, such that

$$x = x_1|| \cdots ||x_\ell||x_*.$$

For this mapping we take $\ell = \lfloor |x|/n \rfloor$, $x_i = x[n(i-1)+1..ni]$ for $i \in [\ell]$, and $x_* = x[n\ell+1..|x|]$. For an incomplete block x, $\mathsf{pad}(x)$ denotes the complete block

$$x||10^* = x||1||0^{n-|x|-1}.$$

For a complete block x, $\mathsf{chop}_k(x)$ denotes the incomplete block $x[1..k]$. For some $m \in \mathbb{N}$, for $x \in \{0,1\}^m, k \in [m]$, $x \gg k$ denotes x rotated b bits to the right, i.e., $0^b||x[1..m-b]$, while $x \ll b$ denotes x rotated b bits to the left, i.e., $x[b+1..m]||0^b$.

We say a function $f : \{0,1\}^n \longrightarrow \{0,1\}^n$ is *partially determined* if we know the values of f on a strict subset of $\{0,1\}^n$. This subset is called $\mathsf{Dom}(f)$. A partially determined state of f can be viewed as a restriction of f to $\mathsf{Dom}(f)$. The range of this restricted function is called $\mathsf{Ran}(f)$. We will treat a partially determined function as *updatable*: for some $x \in \{0,1\}^n \setminus \mathsf{Dom}(f)$ and some $y \in \{0,1\}^n$, (x, y) may be added to f, so that $\mathsf{Dom}(f)$ expands to $\mathsf{Dom}(f) \cup \{x\}$, and $\mathsf{Ran}(f)$ becomes $\mathsf{Ran}(f) \cup \{y\}$. We say f is *permutation-compatible* if $|\mathsf{Dom}(f)| = |\mathsf{Ran}(f)|$.

For a set $\mathcal{S}$, we write $x \xleftarrow{\$} \mathcal{S}$ to denote that x is sampled from S uniformly. For a given domain $\mathcal{D}$ and a given co-domain $\mathcal{R}$, $\mathsf{Func}[\mathcal{D}, \mathcal{R}]$ will denote the set of all functions from $\mathcal{D}$ into $\mathcal{R}$. We say f^* is an ideal random function from $\mathcal{D}$ to $\mathcal{R}$ to indicate that $f^* \xleftarrow{\$} \mathsf{Func}[\mathcal{D}, \mathcal{R}]$. If f^* is an ideal random function, it can be viewed as a *with-replacement* sampler from $\mathcal{R}$: for distinct inputs $x_1, \ldots, x_m \in \mathcal{D}$, $f^*(x_i) \xleftarrow{\$} \mathcal{R}$ for $i \in [m]$, and $f^*(x_1), \ldots, f^*(x_m)$ are all independent. Similarly $\mathsf{Perm}[\mathcal{D}]$ will denote the set of all permutations on $\mathcal{D}$. We say π^* is an ideal random permutation on $\mathcal{D}$ to indicate that $\pi^* \xleftarrow{\$} \mathsf{Perm}[\mathcal{D}]$. If f^* is an ideal random permutation, it can be viewed as a *without-replacement* sampler from $\mathcal{D}$: for distinct inputs $x_1, \ldots, x_m \in \mathcal{D}$, $(\pi^*(x_1), \ldots, \pi^*(x_m)) \xleftarrow{\$} \mathcal{D}^{\underline{s}}$ where $\mathcal{D}^{\underline{s}}$ denotes the set of all s-tuples of distinct elements from $\mathcal{D}$.

2.1 Some Basic Results

We briefly state some results which would be used in our security analysis.

Property-1. Suppose $X_1, \ldots, X_s$ is a random without-replacement sample from $\mathcal{D}$. Then for any $1 \leq i_1 < \cdots < i_r \leq s$, $X_{i_1}, \ldots, X_{i_r}$ is also a random without-replacement sample from $\mathcal{D}$. In other words, the joint distribution of a without-replacement sample is independent of the ordering of the sample.

Property-2. Suppose A is a binary full row rank matrix of dimension $nd \times ns$ for some positive integers n, d and s. Let $X_1, \ldots, X_s$ be a without-replacement sample from $\{0,1\}^n$, and X be the column vector $(X_1, \ldots, X_n)$. Then

$$\Pr[AX = c] \leq \frac{1}{(2^n - s + r) \cdots (2^n - s + 1)}$$

for any d dimensional binary vector c. Moreover if c is not in the column space of A then this probability is zero.

2.2 Distinguishing Advantage

For two oracles $\mathcal{O}_0$ and $\mathcal{O}_1$, an algorithm $\mathcal{A}$ trying to distinguish between $\mathcal{O}_0$ and $\mathcal{O}_1$ is called a distinguishing adversary. $\mathcal{A}$ plays an interactive game with $\mathcal{O}_b$ for some bit b unknown to $\mathcal{A}$, and then outputs a bit $b_{\mathcal{A}}$. The winning event is $[b_{\mathcal{A}} = b]$. The distinguishing advantage of $\mathcal{A}$ is defined as

$$\mathbf{Adv}_{\mathcal{O}_1,\mathcal{O}_0}(\mathcal{A}) := \Big| \Pr[b_{\mathcal{A}} = 1 \mid b = 1] - \Pr[b_{\mathcal{A}} = 1 \mid b = 0] \Big|.$$

Let $\mathbf{A}[q,t]$ be the class of all distinguishing adversaries limited to q oracle queries and t computations. We define

$$\mathbf{Adv}_{\mathcal{O}_1,\mathcal{O}_0}[q,t] := \max_{\mathcal{A} \in \mathbf{A}[q,t]} \mathbf{Adv}_{\mathcal{O}_1,\mathcal{O}_0}(\mathcal{A}).$$

When the adversaries in $\mathbf{A}[q,t]$ are allowed to make both encryption queries and decryption queries to the oracle, this is written as $\mathbf{Adv}_{\pm\mathcal{O}_0,\pm\mathcal{O}_1}[q,q',t]$, where q is the maximum number of encryption queries allowed and q' is the maximum number of decryption queries allowed. Enc_b and Dec_b denote respectively the encryption and decryption function associated with $\mathcal{O}_b$.

$\mathcal{O}_0$ conventionally represents an ideal primitive, while $\mathcal{O}_1$ represents either an actual construction or a mode of operation built of some other ideal primitives. Typically the goal of the function represented by $\mathcal{O}_1$ is to emulate the ideal primitive represented by $\mathcal{O}_0$. We use the standard terms *real oracle* and *ideal oracle* for $\mathcal{O}_1$ and $\mathcal{O}_0$ respectively. A security game is a distinguishing game with an optional set of additional restrictions, chosen to reflect the desired security goal. When we talk of distinguishing advantage with a specific security game G in mind, we include G in the superscript, e.g., $\mathbf{Adv}^{G}_{\mathcal{O}_1,\mathcal{O}_0}(\mathcal{A})$.

2.3 The Authenticated Encryption Security Game

A nonce-based authenticated encryption scheme with associated data consists of a key space $\mathcal{K}$, a message space $\mathcal{M}$, a tag space $\mathfrak{T}$, a nonce space $\mathcal{N}$ and an associated data space $\mathfrak{A}$, along with two functions $\mathsf{Enc} : \mathcal{K}\times\mathcal{N}\times\mathfrak{A}\times\mathcal{M} \longrightarrow \mathcal{M}\times\mathfrak{T}$ and $\mathsf{Dec} : \mathcal{K} \times \mathcal{N} \times \mathfrak{A} \times \mathcal{M} \times \mathfrak{T} \longrightarrow \mathcal{M} \cup \{\bot\}$, with the correctness condition that for any $K \in \mathcal{K}, \mathsf{N} \in \mathcal{N}, \mathsf{A} \in \mathfrak{A}, \mathsf{M} \in \mathcal{M}$, we have

$$\mathsf{Dec}(K, \mathsf{N}, \mathsf{A}, \mathsf{Enc}(K, \mathsf{N}, \mathsf{A}, \mathsf{M})) = \mathsf{M}.$$

In addition, in most popular authenticated encryption schemes (including OCB3), the map $p_{\mathcal{M}} \circ \mathsf{Enc}(K, \mathsf{N}, \mathsf{A}, \cdot)$ for fixed $K, \mathsf{N}, \mathsf{A}$ is a length-preserving permutation, where $p_{\mathcal{M}} : \mathcal{M} \times \mathfrak{T} \longrightarrow \mathcal{M}$ is the projection on $\mathcal{M}$.

In the nonce-respecting authenticated encryption security game with associated data naead, Enc_1 and Dec_1 of the real oracle are the encryption function $\mathsf{Enc}(K, \cdot, \cdot, \cdot)$ and decryption function $\mathsf{Dec}(K, \cdot, \cdot, \cdot, \cdot)$ respectively of the authenticated encryption scheme under consideration for a key K randomly chosen from $\mathcal{K}$; in the ideal oracle, $\mathsf{Enc}_0 : \mathcal{N} \times \mathfrak{A} \times \mathcal{M} \longrightarrow \mathcal{M} \times \mathfrak{T}$ is an ideal random function from $\mathcal{N} \times \mathfrak{A} \times \mathcal{M}$ to $\mathcal{M} \times \mathfrak{T}$, and $\mathsf{Dec}_0 : \mathcal{N} \times \mathfrak{A} \times \mathcal{M} \times \mathfrak{T} \longrightarrow \mathcal{M} \cup \{\bot\}$ is the constant function that returns $\bot$ irrespective of the input. We henceforth refer to $(\mathsf{Enc}_0, \mathsf{Dec}_0)$ as the *ideal nonce-based authenticated encryption scheme.* The distinguishing adversary operates under the following restrictions:

- no two encryption queries can have the same nonce;
- if an encryption query $(\mathsf{N}, \mathsf{A}, \mathsf{M})$ yields (C, T), a decryption query $(\mathsf{N}, \mathsf{A}, \mathsf{C}, \mathsf{T})$ is not allowed.

The distinguishing advantage of the adversary in the nonce-respecting authenticated encryption security game with associated data will be denoted $\mathbf{Adv}^{\mathsf{naead}}_{\mathcal{O}_1,\mathcal{O}_0}(\mathcal{A})$. Note that security under this formulation covers the two standard security goals of authenticated encryption:

- *(Privacy)* Security against an adversary who tries to distinguish the construction from an ideal prf $f^* : \mathcal{M} \longrightarrow \mathcal{M} \times \mathfrak{T}$, and
- *(Integrity)* Security against an adversary who tries to make a successful forging attempt on the construction.

2.4 Coefficients H Technique

Consider a security game G where the adversary can make both encryption queries and decryption queries. The part of the computation visible to the adversary at the time of choosing its final response is known as a view. This includes the queries and the responses, and may also include any additional information the oracle chooses to reveal to the adversary at the end of the query-response phase of the game. The probability of the security game with an oracle $\mathcal{O}$ resulting in a given view V is known as the interpolation probability of V, denoted $\mathsf{ip}_{\mathcal{O}}[V]$.

Note that for a view to be realised, two things need to happen:

- The adversary needs to make the queries listed in the view;
- The oracle needs to make the corresponding responses.

Of these, the former is deterministic; the latter, probabilistic. Thus when we talk of interpolation probability, we are only concerned with the oracle responses, with the assumption that the adversary's queries are consistent with the view. For any other adversary, the interpolation probability is trivially 0. Thus $\mathsf{ip}_{\mathcal{O}}[V]$ depends only on the oracle $\mathcal{O}$ and the view V and not on the adversary; hence the notation.

We extend the notation of interpolation probability to a set $\mathcal{V}$ of views: $\mathsf{ip}_{\mathcal{O}}[\mathcal{V}]$ denotes the probability that the security game with $\mathcal{O}$ results in a view $V \in \mathcal{V}$. Now we state a theorem, due to Jacques Patarin, known as the Coefficient H Technique.

Theorem 2 (Coefficient H Technique). *[Pat08] Suppose there is a set $\mathcal{V}_{\mathsf{bad}}$ of views satisfying the following:*

- $\mathsf{ip}_{\mathcal{O}_0}[\mathcal{V}_{\mathsf{bad}}] \leq \epsilon_1$*;*
- *For any $V \notin \mathcal{V}_{\mathsf{bad}}$,*

$$\frac{\mathsf{ip}_{\mathcal{O}_1}[V]}{\mathsf{ip}_{\mathcal{O}_0}[V]} \geq 1 - \epsilon_2.$$

Then for an adversary $\mathcal{A}$ trying to distinguish between $\mathcal{O}_1$ and $\mathcal{O}_0$, we have the following bound on its distinguishing advantage:

$$\mathbf{Adv}^{G}_{\pm\mathcal{O}_1,\pm\mathcal{O}_0}(\mathcal{A}) \leq \epsilon_1 + \epsilon_2.$$

3 OCB3 Construction

The $\mathsf{OCB3}$ encryption and decryption algorithms are described in Algorithm 1. We take block length $n = 128$. E_K denotes a call to blockcipher, and $\mathcal{H}_K$ denotes a call to a hash function based on the stretch-then-shift xor-universal hash H_κ, as described below in Subsect. 3.1. Note that they share the same key K. Hashing of associated data A is done through parallel masked calls to E_K, which are added to get the authentication key auth. The message space $\mathcal{M}$ consists of all messages with at least one full block, i.e., all strings of 128 bits or more, and the nonce space N consists of all 128-bit strings whose first 122 bits are not all 0. The message M is encrypted in ECB mode, with masking that incorporates the nonce N. If there is an incomplete block at the end, it is added after a 10^* padding to an encrypted masking key. Finally, a checksum of the message blocks is masked and encrypted through E_K, and auth is added to it to produce the authentication tag T. We ignore here the last step of $\mathsf{OCB3}$, where a tag of a desired length τ is obtained by chopping T as required. A schematic view of the encryption is illustrated in Fig. 1, which treats each masked blockcipher call as a call to a tweakable blockcipher. The masking scheme corresponding to the various tweakable blockcipher calls is given in Table 1. The important thing to note here is that *all the masking coefficients are distinct.*

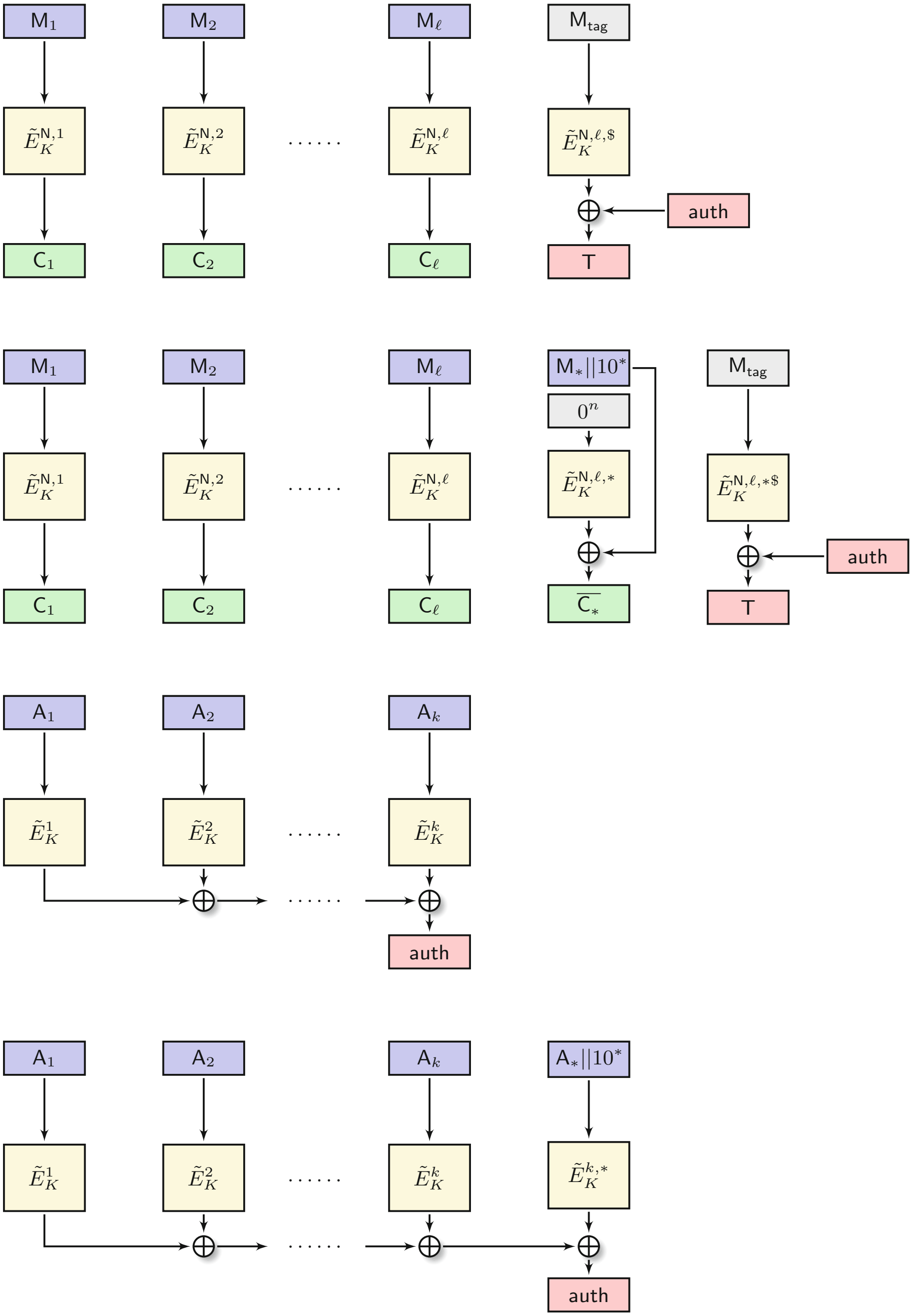

Fig. 1. A schematic view of the OCB3 construction. **Top to Bottom:** encrypting M when $|\mathsf{M}_*| = 0$; encrypting M when $|\mathsf{M}_*| > 0$; hashing A when $|\mathsf{A}_*| = 0$; hashing A when $|\mathsf{A}_*| > 0$.

Table 1. Masking scheme corresponding to the various blockcipher calls in the schematic view of $\mathsf{OCB3}$. $\mathsf{L} = E_K(0)$; $\mathsf{Q} = \mathcal{H}_K(\mathsf{N})$, $\mathcal{H}_K$ being the hash function described in Subsect. 3.1, and N the nonce; and the tweak space is $\mathcal{N} \times ((\mathbb{N} \setminus \{0\}) \cup (\mathbb{N} \times \{*, \$, *\$\})) \cup (\mathbb{N} \setminus \{0\} \cup (\mathbb{N} \times \{*\}))$. For $i \in \mathbb{N}$, $a(i) := \bigoplus_{j \leq i}(1 \ll \mathsf{ntz}(j))$ denotes the Gray-code representation of i, $\mathsf{ntz}(j)$ being the number of trailing 0's in the binary representation of j.

Call	Definition	
$\tilde{E}_K^{\mathsf{N},i}(x)$	$E_K(\mathsf{Q} \oplus \lambda_i \cdot \mathsf{L} \oplus x) \oplus \mathsf{Q} \oplus \lambda_i \cdot \mathsf{L}$,	$i \in \mathbb{N} \setminus \{0\}$
$\tilde{E}_K^{\mathsf{N},i,*}(x)$	$E_K(\mathsf{Q} \oplus \lambda_i^* \cdot \mathsf{L} \oplus x)$,	$i \in \mathbb{N}$
$\tilde{E}_K^{\mathsf{N},i,\$}(x)$	$E_K(\mathsf{Q} \oplus \lambda_i^\$ \cdot \mathsf{L} \oplus x)$,	$i \in \mathbb{N}$
$\tilde{E}_K^{\mathsf{N},i,*\$}(x)$	$E_K(\mathsf{Q} \oplus \lambda_i^{*\$} \cdot \mathsf{L} \oplus x)$,	$i \in \mathbb{N}$
$\tilde{E}_K^{i}(x)$	$E_K(\lambda_i \cdot \mathsf{L} \oplus x)$,	$i \in \mathbb{N} \setminus \{0\}$
$\tilde{E}_K^{i,*}(x)$	$E_K(\lambda_i^* \cdot \mathsf{L} \oplus x)$,	$i \in \mathbb{N}$

Coeff.	Def.
λ_i	$4a(i)$
λ_i^*	$4a(i) + 1$
$\lambda_i^\$$	$4a(i) + 2$
$\lambda_i^{*\$}$	$4a(i) + 3$

3.1 Stretch-then-Shift Hash

In this subsection we describe the hash function $\mathcal{H}_K$ used in $\mathsf{OCB3}$ to process the nonce N. It is based on an xor-universal hash function H_κ with a 128-bit key and a 6-bit input, defined as

$$H_\kappa(x) := ((\kappa || (\kappa \oplus (\kappa \ll 8))) \ll x) \gg 128.$$

This is a linear function of κ, and thus can be described as left multiplication with a matrix $\mathbf{H}[x]$ as

$$H_\kappa(x) := \mathbf{H}[x] \cdot \kappa.$$

It is easy to show that when $\kappa \xleftarrow{\$} \{0,1\}^{128}$, for any $x \in \{0,1\}^6$, we have $H_\kappa(x) \xleftarrow{\$} \{0,1\}^{128}$. The authors show with a computer-aided exhaustive search that when κ is uniform over $\{0,1\}^{128}$, for any $x, x' \in \{0,1\}^6, x \neq x'$ and any $\delta \in \{0,1\}^{128}$, we have

$$\Pr\left[H_\kappa(x) \oplus H_\kappa(x') = \delta\right] = \frac{1}{2^{128}}.$$

We describe here a generalised hash $\mathcal{H}[\pi]$, based on an arbitrary permutation π. We begin by splitting N into two parts:

$$\mathsf{TN} = \tau(\mathsf{N}) := (\mathsf{N} \gg 6) \ll 6,$$
$$\mathsf{BN} = \beta(\mathsf{N}) := \mathsf{N} \oplus \tau(\mathsf{N}),$$

so as BN denotes the last 6 bits of N, and TN denotes the first 122 bits, with 6 0's appended at the end (Note that as long as $\mathsf{N} \in \mathcal{N}$, TN cannot be 0). Next we define

$$\mathsf{KN} = \kappa(\mathsf{N}) := \pi(\tau(\mathsf{N})).$$

Finally, we define

$$\mathsf{Q} = \mathcal{H}[\pi](\mathsf{N}) := H_{\kappa(\mathsf{N})}(\beta(\mathsf{N})) = \mathbf{H}[\beta(\mathsf{N})] \cdot \kappa(\mathsf{N}).$$

The $\mathcal{H}_K$ used in $\mathsf{OCB3}$ is an instantiation $\mathcal{H}[\pi]$ with $\pi = E_K$, i.e.,

$$\mathcal{H}_K(\mathsf{N}) := \mathcal{H}[E_K](\mathsf{N}).$$

4 Security Result

We present the main security result of the paper, along with an overview of our proof approach. Consider a nonce-based authenticated encryption security game with associated data involving $\mathsf{OCB3}[\pi]$, an ideal version of $\mathsf{OCB3}$ where E_K is replaced by a random permutation π. Recall Theorem 1 from Sect. 1.

Theorem 1. *Let $\mathcal{A}$ be an adversary that makes q encryption queries consisting of σ message blocks in all with at most ℓ_{MAX} blocks per query, and α associated data blocks in all, and q' decryption queries in a nonce-respecting authenticated encryption security game with associated data against the oracles $\mathcal{O}_1$ and $\mathcal{O}_0$, where $\mathcal{O}_1$ simulates $\mathsf{OCB3}[\pi]$, and $\mathcal{O}_0$ simulates an ideal nonce-based authenticated encryption scheme with associated data. Then*

$$\mathbf{Adv}^{\mathsf{naead}}_{\mathcal{O}_1,\mathcal{O}_0}(\mathcal{A}) \leq \frac{5\sigma_T^2}{N} + \frac{2\sigma^4}{N^2} + \frac{64q'\ell_{MAX} + 15q'}{N},$$

where $\sigma_T = \sigma + \alpha + q$ is the total number of blocks queried in the encryption queries (including messages, associated data and nonces).

4.1 Proof Approach

Before delving into the details of the proof, we give an overview of it. There are two parts to this security bound: the privacy bound, represented by the term $5\sigma_T^2/N + 2\sigma^4/N^2$, and the integrity bound, represented by the term $(64q'\ell_{\mathrm{MAX}}+15q')/N$. The privacy bound is birthday in the number of encryption-query blocks, and and relies on the simple requirement that every blockcipher output is distinct. The integrity bound, being beyond-birthday in the number of decryption-query blocks (as long as ℓ_{MAX} is within a reasonable bound) is trickier to obtain, and is the main contribution of the paper.

We consider a slightly modified game where we let the real oracle $\mathcal{O}_1$ reveal the inputs and outputs of all internal blockcipher calls in the encryption queries at the end of the query phase. Thus, it becomes necessary for the ideal oracle $\mathcal{O}_0$ to sample these values. In Subsect. 4.3, we describe the sampling order for $\mathcal{O}_0$, which proceeds in four steps. Step 1 takes place during the encryption-query phase itself, when $\mathcal{O}_0$ behaves as in a standard naead game, sampling the ciphertext and tag blocks on the fly. (In the decryption query phase, $\mathcal{O}_0$ always outputs $\perp$.) After the query phase, in Step 2 and Step 3, the inputs and

```
Encryption
input  : $\mathsf{N}, \mathsf{A}, \mathsf{M}$
output: $\mathsf{C}, \mathsf{T}$
begin
    $\mathsf{L} \leftarrow E_K(0)$
    $\mathsf{Q} \leftarrow \mathcal{H}_K(\mathsf{N})$
    $\mathsf{auth} \leftarrow 0$
    for $j \leftarrow 1$ to $k$ do
        $\mathsf{U}_j \leftarrow \lambda_j \cdot \mathsf{L} \oplus \mathsf{A}_j$
        $\mathsf{V}_j \leftarrow E_K(\mathsf{U}_j)$
        $\mathsf{auth} \leftarrow \mathsf{auth} \oplus \mathsf{V}_j$
    end for
    if $|A_*| > 0$ then
        $\overline{\mathsf{A}_*} \leftarrow \mathsf{pad}(\mathsf{A}_*)$
        $\mathsf{U}_* \leftarrow \lambda_k^* \cdot \mathsf{L} \oplus \overline{\mathsf{A}_*}$
        $\mathsf{V}_* \leftarrow E_K(\mathsf{U}_*)$
        $\mathsf{auth} \leftarrow \mathsf{auth} \oplus \mathsf{V}_*$
    end if
    $\mathsf{M}_{\mathsf{tag}} \leftarrow 0$
    for $j \leftarrow 1$ to $\ell$ do
        $\mathsf{X}_j \leftarrow \mathsf{Q} \oplus \lambda_j \cdot \mathsf{L} \oplus \mathsf{M}_j$
        $\mathsf{Y}_j \leftarrow E_K(\mathsf{X}_j)$
        $\mathsf{C}_j \leftarrow \mathsf{Q} \oplus \lambda_j \cdot \mathsf{L} \oplus \mathsf{Y}_j$
        $\mathsf{M}_{\mathsf{tag}} \leftarrow \mathsf{M}_{\mathsf{tag}} \oplus \mathsf{M}_j$
    end for
    $\lambda^i_{\mathsf{tag}} \leftarrow \lambda^{\$}_{\ell}$
    if $|M_*| > 0$ then
        $\overline{\mathsf{M}_*} \leftarrow \mathsf{pad}(\mathsf{M}_*)$
        $\mathsf{X}_* \leftarrow \mathsf{Q} \oplus \lambda^*_{\ell} \cdot \mathsf{L}$
        $\mathsf{Y}_* \leftarrow E_K(\mathsf{X}_*)$
        $\overline{\mathsf{C}_*} \leftarrow \mathsf{Y}_* \oplus \overline{\mathsf{M}_*}$
        $\mathsf{C}_* \leftarrow \overline{\mathsf{C}_*} \gg (n - |\mathsf{M}_*|)$
        $\mathsf{M}_{\mathsf{tag}} \leftarrow \mathsf{M}_{\mathsf{tag}} \oplus \overline{\mathsf{M}_*}$
        $\lambda^i_{\mathsf{tag}} \leftarrow \lambda^{*\$}_{\ell}$
    end if
    $\mathsf{X}_{\mathsf{tag}} \leftarrow \mathsf{Q} \oplus \lambda^i_{\mathsf{tag}} \cdot \mathsf{L} \oplus \mathsf{M}_{\mathsf{tag}}$
    $\mathsf{Y}_{\mathsf{tag}} \leftarrow E_K(\mathsf{X}_{\mathsf{tag}})$
    $\mathsf{T} \leftarrow \mathsf{auth} \oplus \mathsf{Y}_{\mathsf{tag}}$
    return $\mathsf{C}$
    return $\mathsf{T}$
end
```

```
Decryption
input  : $\mathsf{N'}, \mathsf{A'}, \mathsf{C'}, \mathsf{T'}$
output: $\mathsf{M'}$ or $\perp$
begin
    $\mathsf{L'} \leftarrow E_K(0)$
    $\mathsf{Q'} \leftarrow \mathcal{H}_K(\mathsf{N'})$
    $\mathsf{auth'} \leftarrow 0$
    for $j \leftarrow 1$ to $k'$ do
        $\mathsf{U'}_j \leftarrow \lambda_j \cdot \mathsf{L'} \oplus \mathsf{A'}_j$
        $\mathsf{V'}_j \leftarrow E_K(\mathsf{U'}_j)$
        $\mathsf{auth'} \leftarrow \mathsf{auth'} \oplus \mathsf{V'}_j$
    end for
    if $|A'_*| > 0$ then
        $\overline{\mathsf{A}_*} \leftarrow \mathsf{pad}(\mathsf{A'}_*)$
        $\mathsf{U'}_* \leftarrow \lambda^*_{k'} \cdot \mathsf{L'} \oplus \overline{\mathsf{A}_*}$
        $\mathsf{V'}_* \leftarrow E_K(\mathsf{U'}_*)$
        $\mathsf{auth'} \leftarrow \mathsf{auth'} \oplus \mathsf{V'}_*$
    end if
    $\mathsf{M'}_{\mathsf{tag}} \leftarrow 0$
    for $j \leftarrow 1$ to $\ell'$ do
        $\mathsf{Y'}_j \leftarrow \mathsf{Q'} \oplus \lambda_j \cdot \mathsf{L'} \oplus \mathsf{C'}_j$
        $\mathsf{X'}_j \leftarrow E_K^{-1}(\mathsf{Y'}_j)$
        $\mathsf{M'}_j \leftarrow \mathsf{Q'} \oplus \lambda_j \cdot \mathsf{L'} \oplus \mathsf{X'}_j$
        $\mathsf{M'}_{\mathsf{tag}} \leftarrow \mathsf{M'}_{\mathsf{tag}} \oplus \mathsf{M'}_j$
    end for
    $\lambda'^i_{\mathsf{tag}} \leftarrow \lambda^{\$}_{\ell'}$
    if $|C'_*| > 0$ then
        $\overline{\mathsf{C'}_*} \leftarrow \mathsf{pad}(\mathsf{C'}_*)$
        $\mathsf{X'}_* \leftarrow \mathsf{Q'} \oplus \lambda^*_{\ell'} \cdot \mathsf{L'}$
        $\mathsf{Y'}_* \leftarrow E_K(\mathsf{X'}_*)$
        $\overline{\mathsf{M'}_*} \leftarrow \mathsf{Y'}_* \oplus \overline{\mathsf{C'}_*}$
        $\mathsf{M'}_* \leftarrow \overline{\mathsf{M'}_*} \gg (n - |\mathsf{C'}_*|)$
        $\mathsf{M'}_{\mathsf{tag}} \leftarrow \mathsf{M'}_{\mathsf{tag}} \oplus \mathsf{pad}(\mathsf{M'}_*)$
        $\lambda'^i_{\mathsf{tag}} \leftarrow \lambda^{*\$}_{\ell'}$
    end if
    $\mathsf{Y'}_{\mathsf{tag}} \leftarrow \mathsf{auth'} \oplus \mathsf{T'}$
    $\mathsf{X'}_{\mathsf{tag}} \leftarrow E_K^{-1}(\mathsf{Y'}_{\mathsf{tag}})$
    $\mathsf{M''}_{\mathsf{tag}} \leftarrow \mathsf{Q'} \oplus \lambda^i_{\mathsf{tag}} \cdot \mathsf{L'} \oplus \mathsf{X'}_{\mathsf{tag}}$
    if $M'_{tag} = M''_{tag}$ then
        return $\mathsf{M'}$
    else
        return $\perp$
    end if
end
```

Algorithm 1. The $\mathsf{OCB3}$ algorithm. $\mathsf{A}_*, \mathsf{M}_*, \mathsf{A'}_*, \mathsf{C'}_*$ are incomplete (possibly empty) blocks at the end of $\mathsf{A}, \mathsf{M}, \mathsf{A'}, \mathsf{C'}$ respectively. The hash function $\mathcal{H}_K$ is described in Subsect. 3.1

outputs of the internal blockcipher calls in all the encryption queries are sampled. Finally, in Step 4, the inputs and outputs of the internal blockcipher calls in the decryption queries which are not yet determined are sampled, completing the sampling process.

During this sampling process, we keep checking the sampled values for various bad events. badA occurs at the end of Step 1 if there are certain undesirable collisions or multicollisions in the sampled ciphertext and tag blocks. badB or badC occurs at the end of Step 2 or Step 3 respectively if there are certain collisions in the inputs or sampled outputs of the internal blockcipher calls. Finally, $\mathsf{badD}[i]$ occurs at the end of Step 4 if after sampling the inputs and outputs of the internal blockcipher calls in the i-th decryption query it turns out that the correct output of $\mathcal{O}_0$ should not have been $\perp$. $\mathsf{badD}[i]$ corresponds to the violation of integrity security, and the bounding of the probability of $\mathsf{badD}[i]$, which is done by carefully selecting the specific collisions we need to ban, forms the heart of this paper.

In Subsect. 4.5, we calculate the probabilities of the various bad cases. The calculations for badA, badB and badC are straightforward. For $\mathsf{badD}[i]$, we look at several cases, and establish a bound for $\mathsf{badD}[i]$ based on some lemmas the proof of which we defer to Sect. 5. By Property-1 in Subsect. 2.1, we can reorder the sampling phase in Step 4 to first sample the blockcipher outputs required for $\mathsf{badD}[i]$. Finally, we bound the probability of $\cup_{i=1}^{q'}\mathsf{badD}[i]$ by the union-bound. In Sect. 5, we prove the lemmas through an exhaustive case-analysis.

4.2 Notation for Adversary Interactions

First we set up the notation for the adversary interactions in the game described in Theorem 1. The i-th encryption query consists of

- a message M^i, consisting of $\ell^i \geq 1$ complete blocks and an incomplete (possibly empty) block M^i_* at the end;
- associated data A^i, consisting of $k^i \geq 1$ blocks and an incomplete (possibly empty) block A^i_* at the end;
- a nonce block N^i, with the first 122 bits not all zero, such that for any $i' \in [i-1]$, $\mathsf{N}^i \neq \mathsf{N}^{i'}$.

Following the notation for $\mathcal{H}[\pi]$ described in Subsect. 3.1, we define $\mathsf{TN}^i := \tau(\mathsf{N}^i), \mathsf{BN}^i := \beta(\mathsf{N}^i), \mathbf{H}^i := \mathbf{H}[\mathsf{BN}^i]$. The corresponding output consists of

- a ciphertext C^i, consisting of ℓ^i complete blocks and an incomplete block C^i_* at the end, with $|\mathsf{C}^i_*| = |\mathsf{M}^i_*|$;
- a tag block T^i.

The i-th decryption query consists of

- a ciphertext $\mathsf{C'}^i$, consisting of $\ell'^i \geq 1$ complete blocks and an incomplete (possibly empty) block $\mathsf{C'}^i_*$ at the end;
- a tag block $\mathsf{T'}^i$;

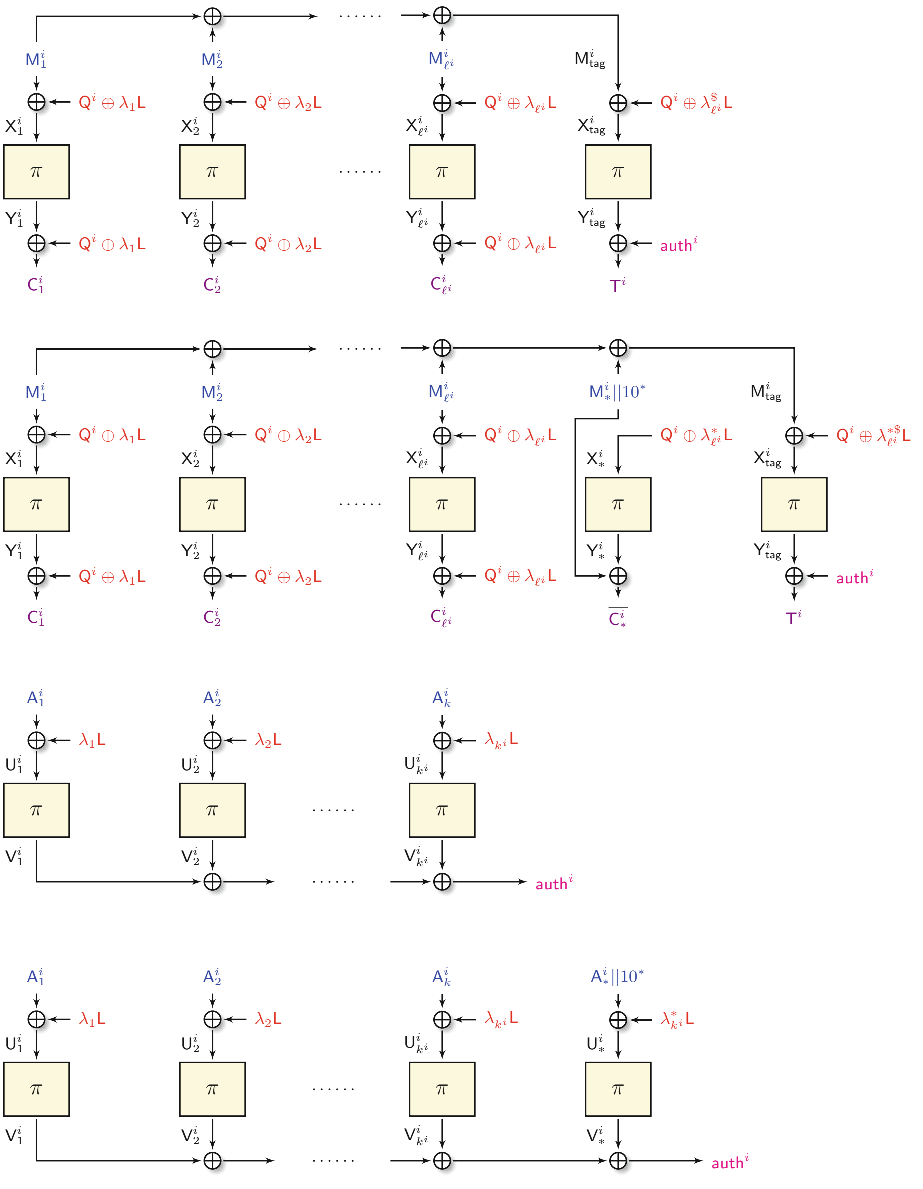

Fig. 2. The OCB3[π] construction: notation for the i-th encryption query. L denotes $\pi(0)$ and Q^i denotes $\mathcal{H}[\pi](N^i)$. **Top to Bottom:** encrypting M^i when $|M_*^i| = 0$; encrypting M^i when $|M_*^i| > 0$; hashing A^i when $|A_*^i| = 0$; hashing A^i when $|A_*^i| > 0$.

- associated date $\mathsf{A'}^i$, consisting of k'^i blocks and an incomplete (possibly empty) block $\mathsf{A'}^i_*$ at the end;
- a nonce block $\mathsf{N'}^i$, with the first 122 bits not all zero.

(Note that in the decryption queries, nonces are allowed to repeat.) Again, as in the i-th encryption query, we define $\mathsf{TN'}^i := \tau(\mathsf{N'}^i), \mathsf{BN'}^i := \beta(\mathsf{N'}^i), \mathbf{H'}^i := \mathbf{H}[\mathsf{BN'}^i]$. The response is either $\perp$, or a message $\mathsf{M'}^i$ consisting of ℓ'^i complete blocks and an incomplete block $\mathsf{M'}^i_*$ at the end, with $|\mathsf{M'}^i_*| = |\mathsf{C'}^i_*|$.

4.3 Oracle Behaviour

Now we describe the oracles involved in the game in greater detail. Let $\mathcal{I}$ (resp. $\mathcal{I}'$) denote the indices for the encryption (resp. decryption) queries with incomplete-block messages, and let $\mathcal{J}$ (resp. $\mathcal{J}'$) denote the indices for the encryption (resp. decryption) queries with incomplete-block associated data. Let

$$\mathcal{F} := \left\{ i \in [q] \mid (\nexists i' < i)(\mathsf{TN}^{i'} = \mathsf{TN}^i) \right\}$$

be the set of first-appearance indices of the distinct values taken by TN^i.

Real Oracle. Enc_1 and Dec_1 of the real oracle $\mathcal{O}_1$ represent the encryption and decryption functions of $\mathsf{OCB3}[\pi]$ respectively. The notation we use for the internal computations of $\mathcal{O}_1$ while responding to the i-th encryption (resp. decryption) query is illustrated in Fig. 2 (resp. Fig. 3). In addition, still following the notation from Subsect. 3.1, for $i \in [q]$ we define $\mathsf{KN}^i := \pi(\mathsf{TN}^i), \mathsf{Q}^i := \mathbf{H}^i \cdot \mathsf{KN}^i$, and for $i \in [q']$ we define $\mathsf{KN'}^i := \pi(\mathsf{TN'}^i), \mathsf{Q'}^i := \mathbf{H'}^i \cdot \mathsf{KN'}^i$. We keep track of $\mathsf{Dom}(\pi)$, the set of inputs to π, and $\mathsf{Ran}(\pi)$, the set of outputs from π. At the end of the query phase, the partially determined π is also revealed to the adversary.

Ideal Oracle. Dec_0 of the ideal oracle is the constant function returning $\perp$. Enc_0 samples and returns $(\mathsf{C}^i, \mathsf{T}^i)$ for the i-th query. At the end of the query phase, the $\mathcal{O}_0$ partially samples π and gives it to the adversary. The sampling behaviour followed by $\mathcal{O}_0$ is described in the subsequent paragraphs. (Note that if one of the bad events badA, badB, badC, or $\mathsf{badD}[i]$ for some $i \in [q']$ is encountered by $\mathcal{O}_0$, its behaviour thereafter is undefined.)

Step 1 and badA. This step is *online*—it takes place during the query phase. For $i \in [q]$, on the i-th encryption query, for each $j \in [\ell^i]$, sample C^i_j uniformly with replacement from $\{0,1\}^n$ and return C^i_j to the adversary; sample T^i uniformly with replacement from $\{0,1\}^n$ and return T^i to the adversary; and if $i \in \mathcal{I}$, sample $\overline{\mathsf{C}^i_*}$ uniformly with replacement from $\{0,1\}^n$, set $\mathsf{C}^i_* = \mathsf{chop}_{|\mathsf{M}^i_*|}(\overline{\mathsf{C}^i_*})$; and return C^i_* to the adversary.

badA occurs when we have

$$\mathsf{C}^{i_1}_{j_1} + \mathsf{C}^{i_1}_{j'_1} = \mathsf{C}^{i_2}_{j_2} + \mathsf{C}^{i_2}_{j'_2} = \mathsf{C}^{i_3}_{j_2} + \mathsf{C}^{i_3}_{j'_2}$$

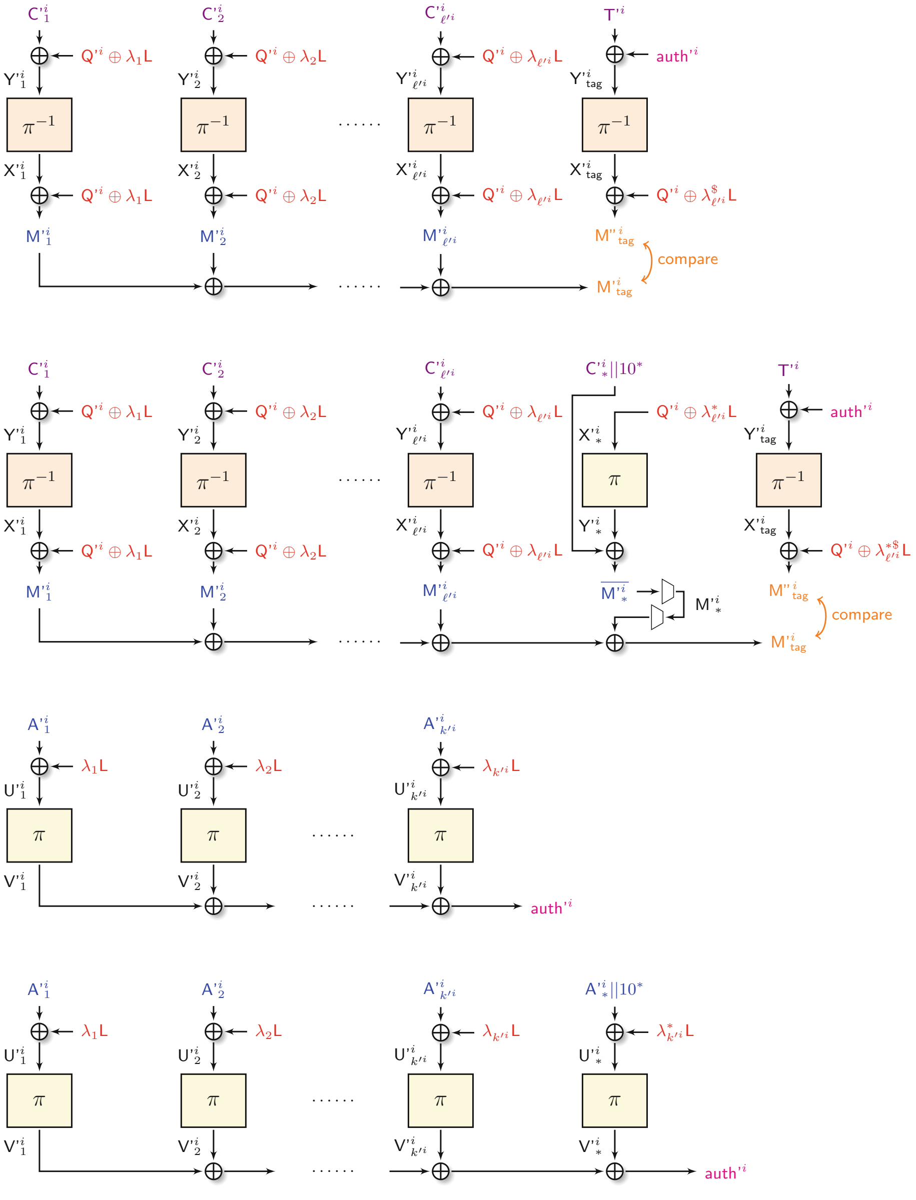
$\mathsf{C'}^i_1$
$\mathsf{C'}^i_2$
$\mathsf{C'}^i_{\ell'^i}$
$\mathsf{T'}^i$
$\mathsf{Q'}^i \oplus \lambda_1 \mathsf{L}$
$\mathsf{Q'}^i \oplus \lambda_2 \mathsf{L}$
$\mathsf{Q'}^i \oplus \lambda_{\ell'^i} \mathsf{L}$
$\mathsf{auth'}^i$
$\mathsf{Y'}^i_1$
$\mathsf{Y'}^i_2$
$\mathsf{Y'}^i_{\ell'^i}$
$\mathsf{Y'}^i_{\mathsf{tag}}$
π^{-1}
......
$\mathsf{X'}^i_1$
$\mathsf{X'}^i_2$
$\mathsf{X'}^i_{\ell'^i}$
$\mathsf{X'}^i_{\mathsf{tag}}$
$\mathsf{Q'}^i \oplus \lambda^{\$}_{\ell'^i} \mathsf{L}$
$\mathsf{M'}^i_1$
$\mathsf{M'}^i_2$
$\mathsf{M'}^i_{\ell'^i}$
$\mathsf{M''}^i_{\mathsf{tag}}$
compare
$\mathsf{M'}^i_{\mathsf{tag}}$
$\mathsf{C'}^i_* || 10^*$
$\mathsf{Q'}^i \oplus \lambda^*_{\ell'^i} \mathsf{L}$
$\mathsf{X'}^i_*$
π
$\mathsf{Y'}^i_*$
$\mathsf{Q'}^i \oplus \lambda^{*\$}_{\ell'^i} \mathsf{L}$
$\overline{\mathsf{M'}^i_*}$
$\mathsf{M'}^i_*$
$\mathsf{A'}^i_1$
$\mathsf{A'}^i_2$
$\mathsf{A'}^i_{k'^i}$
$\lambda_1 \mathsf{L}$
$\lambda_2 \mathsf{L}$
$\lambda_{k'^i} \mathsf{L}$
$\mathsf{U'}^i_1$
$\mathsf{U'}^i_2$
$\mathsf{U'}^i_{k'^i}$
$\mathsf{V'}^i_1$
$\mathsf{V'}^i_2$
$\mathsf{V'}^i_{k'^i}$
$\mathsf{A'}^i_* || 10^*$
$\lambda^*_{k'^i} \mathsf{L}$
$\mathsf{U'}^i_*$
$\mathsf{V'}^i_*$

Fig. 3. The $\mathsf{OCB3}[\pi]$ construction: notation for the i-th decryption query. L denotes $\pi(0)$ and $\mathsf{Q'}^i$ denotes $\mathcal{H}[\pi](\mathsf{N'}^i)$. **Top to Bottom:** decrypting $\mathsf{C'}^i$ when $|\mathsf{C'}^i_*| = 0$; decrypting $\mathsf{C'}^i$ when $|\mathsf{C'}^i_*| > 0$; hashing $\mathsf{A'}^i$ when $|\mathsf{A'}^i_*| = 0$; hashing $\mathsf{A'}^i$ when $|\mathsf{A'}^i_*| > 0$.

for some $i_1, i_2, i_3 \in [q]$ and three distinct pairs $(j_1, j_1'), (j_2, j_2'), (j_3, j_3')$ satisfying

$$\lambda_{j_1} + \lambda_{j_1'} = \lambda_{j_2} + \lambda_{j_2'} = \lambda_{j_3} + \lambda_{j_3'}.$$

This restriction on certain multi-collisions over the ciphertexts is required in the proof of Lemma 5 in Sect. 5. The remaining steps of the simulation take place after the query phase is over.

Step 2 and badB. Begin with $\pi = \{\}$ (so that $\mathsf{Dom}(\pi) = \mathsf{Ran}(\pi) = \{\}$). Sample L uniformly from $\{0,1\}^n$. For $i \in \mathcal{F}$, sample KN^i uniformly without replacement from $\{0,1\}^n \setminus \{\mathsf{L}\}$. Next set the following values:

- for $i \in [q]$, set $\mathsf{Q}^i = \mathbf{H}^i \cdot \mathsf{KN}^i$;
- for $i \in [q], j \in [\ell^i]$ set $\mathsf{X}^i_j = \mathsf{M}^i_j + \mathsf{Q}^i + \lambda_j \cdot \mathsf{L}$ and $\mathsf{Y}^i_j = \mathsf{C}^i_j + \mathsf{Q}^i + \lambda_j \cdot \mathsf{L}$;
- for $i \in \mathcal{I}$ set $\mathsf{X}^i_* = \mathsf{Q}^i + \lambda^*_{\ell^i} \cdot \mathsf{L}$ and $\mathsf{Y}^i_* = \mathsf{M}^i_* || 10^* + \overline{\mathsf{C}^i_*}$;
- for $i \in [q] \setminus \mathcal{I}$ set $\mathsf{M}^i_{\mathsf{tag}} = \sum_{i=1}^{\ell^i} \mathsf{M}^i_j$ and $\mathsf{X}^i_{\mathsf{tag}} = \mathsf{M}^i_{\mathsf{tag}} + \mathsf{Q}^i + \lambda^{\$}_{\ell^i} \cdot \mathsf{L}$;
- for $i \in \mathcal{I}$ set $\mathsf{M}^i_{\mathsf{tag}} = \sum_{i=1}^{\ell^i} \mathsf{M}^i_j + \mathsf{M}^i_*$ and $\mathsf{X}^i_{\mathsf{tag}} = \mathsf{M}^i_{\mathsf{tag}} + \mathsf{Q}^i + \lambda^{*\$}_{\ell^i} \cdot \mathsf{L}$;
- for $i \in [q], j \in [k^i]$ set $\mathsf{U}^i_j = \mathsf{A}^i_j + \lambda_j \cdot \mathsf{L}$;
- for $i \in \mathcal{J}$ set $\mathsf{U}^i_* = \mathsf{A}^i_* || 10^* + \lambda^*_{k^i} \cdot \mathsf{L}$.

badB occurs when:

- there are collisions in the values 0, TN^i for $i \in \mathcal{F}$, X^i_j for $i \in [q], j \in [\ell^i]$, X^i_* for $i \in \mathcal{I}$, $\mathsf{X}^i_{\mathsf{tag}}$ for $i \in [q]$, U^i_j for $i \in [q], j \in [k^i]$, U^i_* for $i \in \mathcal{J}$, not counting the trivial collisions $\mathsf{U}^i_j = \mathsf{U}^{i'}_j$ when $\mathsf{A}^i_j = \mathsf{A}^{i'}_j$; or
- there are collisions in the values L, KN^i for $i \in \mathcal{F}$, Y^i_j for $i \in [q], j \in [\ell^i]$, Y^i_* for $i \in \mathcal{I}$.

Add the following to π:

- $(0, \mathsf{L})$;
- $(\mathsf{TN}^i, \mathsf{KN}^i)$ for $i \in \mathcal{F}$;
- $(\mathsf{X}^i_j, \mathsf{Y}^i_j)$ for $i \in [q], j \in [\ell^i]$;
- $(\mathsf{X}^i_*, \mathsf{Y}^i_*)$ for $i \in \mathcal{I}$;

Note that the π sampled thus far remains permutation-compatible as long as badB does not occur.

Step 3 and badC. For each distinct $\mathsf{U}^i_j, i \in [q], j \in [k^i]$, sample V^i_j uniformly without replacement from $\{0,1\}^n \setminus \mathsf{Ran}(\pi)$. For each distinct U^i_* for $i \in \mathcal{J}$, sample V^i_* uniformly without replacement from $\{0,1\}^n \setminus (\mathsf{Ran}(\pi) \cup \{\mathsf{V}^i_j \mid j \in [k^i]\})$. Next set the following values:

- for $i \in [q] \setminus \mathcal{J}$ set $\mathsf{auth}^i = \sum_{j=1}^{k^i} \mathsf{V}^i_j$;
- for $i \in \mathcal{J}$ set $\mathsf{auth}^i = \sum_{j=1}^{k^i} \mathsf{V}^i_j + \mathsf{V}^i_*$;
- for $i \in [q]$ set $\mathsf{Y}^i_{\mathsf{tag}} = \mathsf{T}^i + \mathsf{auth}^i$.

badC occurs when:

- $\mathsf{Y}^i_{\mathsf{tag}} \in \mathsf{Ran}(\pi)$ for some $i \in [q]$;
- $\mathsf{Y}^i_{\mathsf{tag}} = \mathsf{V}^i_j$ for some $i \in [q], j \in [k^i]$;
- $\mathsf{Y}^i_{\mathsf{tag}} = \mathsf{V}^i_*$ for some $i \in \mathcal{I}$; or
- $\mathsf{Y}^i_{\mathsf{tag}} = \mathsf{Y}^{i'}_{\mathsf{tag}}$ for some $i, i' \in [q]$.

Add the following to π:

- $(\mathsf{X}^i_{\mathsf{tag}}, \mathsf{Y}^i_{\mathsf{tag}})$ for $i \in [q]$;
- $(\mathsf{U}^i_j, \mathsf{V}^i_j)$ for $i \in [q], j \in [k^i]$;
- $(\mathsf{U}^i_*, \mathsf{V}^i_*)$ for $i \in \mathcal{J}$;

Note that the π sampled thus far remains permutation-compatible as long as neither of badB and badC occurs.

Step 4 and $\mathsf{badD}[i]$. In this step we keep updating π (and hence $\mathsf{Dom}(\pi)$ and $\mathsf{Ran}(\pi)$) on the fly. For each $i \in [q']$, set $\mathsf{M'}^i_{\mathsf{tag}}$ and $\mathsf{M''}^i_{\mathsf{tag}}$ as follows:

- If $\mathsf{TN'}^i \in \mathsf{Dom}(\pi)$, set $\mathsf{KN'}^i = \pi(\mathsf{TN'}^i)$, otherwise sample $\mathsf{KN'}^i$ uniformly without replacement from $\{0,1\}^n \setminus \mathsf{Ran}(\pi)$, and add $(\mathsf{TN'}^i, \mathsf{KN'}^i)$ to π;
- For $j \in [\ell'^i]$, set $\mathsf{Y'}^i_j = \mathsf{C'}^i_j + \mathsf{Q'}^i + \lambda_j \cdot \mathsf{L}$; if $\mathsf{Y'}^i_j \in \mathsf{Ran}(\pi)$, set $\mathsf{X'}^i_j = \pi^{-1}(\mathsf{Y'}^i_j)$, otherwise sample $\mathsf{X'}^i_j$ uniformly without replacement from $\{0,1\}^n \setminus \mathsf{Dom}(\pi)$, and add $(\mathsf{X'}^i_j, \mathsf{Y'}^i_j)$ to π; finally, set $\mathsf{M'}^i_j = \mathsf{X'}^i_j + \mathsf{Q'}^i + \lambda_j \cdot \mathsf{L}$;
- If $i \in \mathcal{I}'$, set $\mathsf{X}^i_* = \mathsf{Q'}^i + \lambda^*_{\ell'^i} \cdot \mathsf{L}$; if $\mathsf{X}^i_* \in \mathsf{Dom}(\pi)$, set $\mathsf{Y}^i_* = \pi(\mathsf{X}^i_*)$, otherwise sample Y^i_* uniformly without replacement from $\{0,1\}^n \setminus \mathsf{Ran}(\pi)$, and add $(\mathsf{X}^i_*, \mathsf{Y}^i_*)$ to π; finally, set $\overline{\mathsf{M'}^i_*} = \mathsf{C'}^i_* || 10^* + \mathsf{Y}^i_*$;
- If $i \notin \mathcal{I}'$, set $\mathsf{M'}^i_{\mathsf{tag}} = \sum_{j=1}^{\ell'^i} \mathsf{M'}^i_j$;
- If $i \in \mathcal{I}'$, set $\mathsf{M'}^i_{\mathsf{tag}} = \sum_{j=1}^{\ell'^i} \mathsf{M'}^i_j + \mathsf{M'}^i_*$;
- For $j \in [k'^i]$, set $\mathsf{U'}^i_j = \mathsf{A'}^i_j + \lambda_j \cdot \mathsf{L}$; if $\mathsf{U'}^i_j \in \mathsf{Dom}(\pi)$, set $\mathsf{V'}^i_j = \pi(\mathsf{U'}^i_j)$, otherwise sample $\mathsf{V'}^i_j$ uniformly without replacement from $\{0,1\}^n \setminus \mathsf{Ran}(\pi)$, and add $(\mathsf{U'}^i_j, \mathsf{V'}^i_j)$ to π;
- If $i \in \mathcal{J}'$, set $\mathsf{U}^i_* = \mathsf{A}^i_* || 10^* + \lambda^*_{\ell'^i} \cdot \mathsf{L}$; if $\mathsf{U}^i_* \in \mathsf{Dom}(\pi)$, set $\mathsf{V}^i_* = \pi(\mathsf{U}^i_*)$, otherwise sample V^i_* uniformly without replacement from $\{0,1\}^n \setminus \mathsf{Ran}(\pi)$, and add $(\mathsf{U}^i_*, \mathsf{V}^i_*)$ to π;
- If $i \notin \mathcal{J}'$, set $\mathsf{auth'}^i = \sum_{j=1}^{k'^i} \mathsf{V'}^i_j$;
- If $i \in \mathcal{J}'$, set $\mathsf{auth'}^i = \sum_{j=1}^{k'^i} \mathsf{V'}^i_j + \mathsf{V'}^i_*$;
- Set $\mathsf{Y'}^i_{\mathsf{tag}} = \mathsf{T'}^i + \mathsf{auth'}^i$; if $\mathsf{Y'}^i_{\mathsf{tag}} \in \mathsf{Ran}(\pi)$, set $\mathsf{X'}^i_{\mathsf{tag}} = \pi^{-1}(\mathsf{Y'}^i_{\mathsf{tag}})$, otherwise sample $\mathsf{X'}^i_{\mathsf{tag}}$ uniformly without replacement from $\{0,1\}^n \setminus \mathsf{Dom}(\pi)$, and add $(\mathsf{Y'}^i_{\mathsf{tag}}, \mathsf{X'}^i_{\mathsf{tag}})$ to π; finally, set $\mathsf{M''}^i_{\mathsf{tag}} = \mathsf{X'}^i_{\mathsf{tag}} + \mathsf{Q'}^i + \lambda^{*\$}_{\ell'^i} \cdot \mathsf{L}$;
- Return π to the adversary.

$\mathsf{badD}[i]$ occurs when $\mathsf{M'}^i_{\mathsf{tag}} = \mathsf{M''}^i_{\mathsf{tag}}$.

4.4 Notation for the Proof

Before we begin the proof, we introduce some more notation. Let $\mathcal{P}'^i$ denote the set of positions for the i-th decryption query, defined as

$$\mathcal{P}'^i := \begin{cases} [\ell'^i] \cup \{\mathsf{tag}\}\,, & i \notin \mathcal{I}', \\ [\ell'^i] \cup \{*, \mathsf{tag}\}\,, & i \in \mathcal{I}'. \end{cases}$$

Further, let $\mathcal{P}'^i(-) := \mathcal{P}'^i \setminus \{\mathsf{tag}\}$. For $p \in \mathcal{P}'^i$, let Δ'^i_p denote the masking key for position p in i-th decryption query, defined as

$$\Delta'^i_p := \begin{cases} \lambda_p \cdot \mathsf{L} + \mathsf{Q}'^i, & p \in [\ell'^i], \\ \lambda'^i_{\mathsf{tag}} \cdot \mathsf{L} + \mathsf{Q}'^i, & p = \mathsf{tag}, \\ \lambda^*_{\ell'^i} \cdot \mathsf{L} + \mathsf{Q}'^i, & p = *, i \in \mathcal{I}', \end{cases}$$

where

$$\lambda'^i_{\mathsf{tag}} := \begin{cases} \lambda^{\$}_{\ell'^i}, & i \notin \mathcal{I}', \\ \lambda^{*\$}_{\ell'^i}, & i \in \mathcal{I}'. \end{cases}$$

For convenience, we will abuse the notation of set membership and extend it to sequences. Thus, for a sequence S and a block Z, $\mathsf{Z} \in \mathsf{S}$ will imply that Z occurs somewhere in S.

4.5 Proof of Theorem

Let $\mathcal{V}_{\mathsf{bad}}$ consist of all those transcripts where one of badA, badB, badC or $\mathsf{badD}[i]$ for some $i \in [q']$ has been encountered. Then

$$\mathsf{ip}_{\mathcal{O}_0}[\mathcal{V}_{\mathsf{bad}}] \leq \Pr_{\mathcal{O}_0}[\mathsf{badA}] + \Pr_{\mathcal{O}_0}[\mathsf{badB}] + \Pr_{\mathcal{O}_0}[\mathsf{badC}] + \sum_{i=1}^{q'} \Pr_{\mathcal{O}_0}[\mathsf{badD}[i]]\,.$$

We make the following claim:

Claim. We have the following bounds on the bad events under $\mathcal{O}_0$:

$$\Pr_{\mathcal{O}_0}[\mathsf{badA}] \leq \frac{\sigma^4}{N^2}, \tag{1}$$

$$\Pr_{\mathcal{O}_0}[\mathsf{badB}] \leq \frac{3\sigma_T^2}{N}, \tag{2}$$

$$\Pr_{\mathcal{O}_0}[\mathsf{badC}] \leq \frac{2q\sigma_T}{N}, \tag{3}$$

$$\Pr_{\mathcal{O}_0}[\mathsf{badD}[i]] \leq \frac{64\ell_{\mathrm{MAX}} + 15}{N} + \frac{32\sigma_T^2}{N^2}. \tag{4}$$

From the claim we have

$$\mathsf{ip}_{\mathcal{O}_0}[\mathcal{V}_{\mathsf{bad}}] \leq \frac{\sigma^4}{N^2} + \frac{3\sigma_T^2}{N} + \frac{2q\sigma_T}{N} + \sum_{i=1}^{q'} \left(\frac{64\ell_{\mathrm{MAX}} + 15}{N} + \frac{32\sigma_T^2}{N^2} \right)$$
$$\leq \frac{5\sigma_T^2}{N} + \frac{2\sigma^4}{N^2} + \frac{64q'\ell_{\mathrm{MAX}} + 15q'}{N}.$$

Consider a view $V \notin \mathcal{V}_{\mathsf{bad}}$. In the real oracle, to obtain V, exactly $\sigma_T + |\mathcal{F}| + 1$ calls are made to π: one for each message block, one for each position-wise distinct associated data block, one for each distinct TN^i, one for 0, and one for each tag. We know that these are all distinct because neither of badB and badC has been encountered. Hence

$$\mathsf{ip}_{\mathcal{O}_1}[V] = \frac{1}{N^{\underline{\sigma_T + |\mathcal{F}| + 1}}}.$$

In the ideal oracle, in Step 1, the $\sigma + q$ online outputs are sampled uniformly with replacement. In Steps 2 and 3, $|\mathcal{F}| + 1 + \alpha$ outputs are sampled uniformly without replacement. Finally, since $\mathsf{badD}[i]$ was not encountered for any $i \in [q']$, all decryption queries in V must have returned $\perp$, which $\mathcal{O}_0$ always returns. Hence

$$\mathsf{ip}_{\mathcal{O}_0}[V] = \frac{1}{N^{\sigma+q}} \cdot \frac{1}{N^{\underline{|\mathcal{F}|+1+\alpha}}} \leq \frac{1}{N^{\underline{\sigma_T + |\mathcal{F}| + 1}}} = \mathsf{ip}_{\mathcal{O}_1}[V].$$

Theorem 2 then gives us the required result. □

Proof of Claim. Suppose badA is encountered. Then we have

$$\mathsf{C}^{i_1}_{j_1} + \mathsf{C}^{i_1}_{j'_1} = \mathsf{C}^{i_2}_{j_2} + \mathsf{C}^{i_2}_{j'_2} = \mathsf{C}^{i_3}_{j_2} + \mathsf{C}^{i_3}_{j'_2},$$
$$\lambda_{j_1} + \lambda_{j'_1} = \lambda_{j_2} + \lambda_{j'_2} = \lambda_{j_3} + \lambda_{j'_3},$$

for some $i_1, i_2, i_3 \in [q]$ and three distinct pairs $(j_1, j'_1), (j_2, j'_2), (j_3, j'_3)$. Now for fixed $i_1, i_2, i_3, j_1, j'_1, j_2, j'_2, j_3, j'_3$, this probability is at most $1/N^2$. Notice that if for any choice of (j_1, j_2, j_3, j'_1), there is at most one choice of j'_2 and at most one choice of j'_3. For any choice of i_1, there are at most σ^2 choices for (i_2, j_2) and (i_3, j_3), and at most $(\ell^i)^2$ choices for (j_1, j'_1). Summing over i gives us

$$\Pr_{\mathcal{O}_0}[\mathsf{badA}] \leq \frac{\sigma^2}{N^2} \sum_{i=1}^{q} (\ell^i)^2 \leq \frac{\sigma^4}{N^2},$$

establishing (1). For badB, since we are now sampling without replacement, each collision event has probability at most $1/(N-1)$. There are at most $(q+1)$ values among 0, TN^i for $i \in \mathcal{F}$, and they are all distinct by sampling; there are at most α distinct values among U^i_j for $i \in [q], j \in [k^i]$, U^i_* for $i \in \mathcal{J}$; and there are $\sigma + q$ values among X^i_j for $i \in [q], j \in [\ell^i]$, X^i_* for $i \in \mathcal{I}$ and $\mathsf{X}^i_{\mathsf{tag}}$ for $i \in [q]$. These give us at most $(q+1)(\sigma + \alpha + q) + (\sigma + \alpha + q)^2/2$ possible collision pairs. Similarly, among L, KN^i for $i \in \mathcal{F}$, Y_j^i for $i \in [q], j \in [\ell^i]$, and Y_*^i for $i \in \mathcal{I}$, there are at most $(q+1)\sigma + \sigma^2/2$ possible collision pairs. Thus we have

$$\begin{aligned}\Pr_{\mathcal{O}_0}[\mathsf{badB}] &\le \frac{(q+1)(\sigma+\alpha+q)}{N-1} + \frac{(\sigma+\alpha+q)^2}{2(N-1)} + \frac{(q+1)\sigma}{N-1} + \frac{\sigma^2}{2(N-1)} \\ &\le \frac{3(\sigma+\alpha+q)^2}{N} = \frac{3\sigma_T^2}{N},\end{aligned}$$

establishing (2). For badC, since at this point $|\mathsf{Ran}(\pi)| = \rho := \sigma + |\mathcal{F}| + 1$, the probability of each collision event is at most $1/(N-\rho-1)$. Since there are at most $q\rho + q\alpha + q^2/2$ possible collision pairs, we have

$$\Pr_{\mathcal{O}_0}[\mathsf{badC}] \le \frac{q\rho}{N-\rho-1} + \frac{q\alpha}{N-\rho-1} + \frac{q^2}{2(N-\rho-1)} \le \frac{2q\sigma_T}{N},$$

establishing (3). To prove (4) we need to bound the probability of $\mathsf{badD}[i]$, and for that we consider several cases. For now we assume that $i \in \mathcal{I}'$. Then $\mathcal{P}'^i(-)$ is simply $[\ell'^i]$. For some $p \in \mathcal{P}'^i$, we say $\mathsf{X'}_p^i$ is *trivially determined* if the adversary can deduce the value of $\mathsf{X'}_p^i$ from the transcript of the encryption queries. This can happen in two ways:

- When $p \in [\ell'^i]$, $\mathsf{X'}_p^i$ is trivially determined if for some i' we have $\mathsf{N'}^i = \mathsf{N}^{i'}$ and $\mathsf{C'}_p^i = \mathsf{C}_p^{i'}$, which forces $\mathsf{X'}_p^i$ to equal $\mathsf{X}_p^{i'}$ — then we say $\mathsf{X'}_p^i$ is i'-trivial;
- $\mathsf{X'}_{\mathsf{tag}}^i$ is trivially determined if for some j we have $\mathsf{A'}^i = \mathsf{A}^j$ and $\mathsf{T'}^i = \mathsf{T}^j$, which forces $\mathsf{X'}_{\mathsf{tag}}^i$ to equal $\mathsf{X}_{\mathsf{tag}}^j$ — then we say $\mathsf{X'}_{\mathsf{tag}}^i$ is j-trivial.

We look at five cases:

- **Case 1.** $\mathsf{X'}_p^i$ is trivially determined for all $p \in \mathcal{P}'^i$;
- **Cases 2 and 3.** For some $p_0 \in \mathcal{P}'^i$, $\mathsf{X'}_{p_0}^i$ is not trivially determined, and $\mathsf{X'}_p^i$ is trivially determined for all $p \in \mathcal{P}'^i \setminus \{p_0\}$:
  - **Case 2.** $p_0 \in [\ell'^i]$;
  - **Case 3.** $p_0 = \mathsf{tag}$;
- **Case 4.** For some $p_0 \in [\ell'^i]$, $\mathsf{X'}_{p_0}^i$ and $\mathsf{X'}_{\mathsf{tag}}^i$ are not trivially determined, and $\mathsf{X'}_p^i$ is trivially determined for all $p \in [\ell'^i] \setminus \{p_0\}$;
- **Case 5.** For some distinct $p_0, p_1 \in [\ell'^i]$, $\mathsf{X'}_{p_0}^i$ and $\mathsf{X'}_{p_1}^i$ are not trivially determined.

Note that the decryption query satisfies one of the five cases above, and moreover that this case can be chosen in advance by the adversary, by appropriately setting the query parameters. Accordingly, we can divide $[q']$ into five disjoint subsets $\mathcal{S}[1], \ldots, \mathcal{S}[5]$, such that for $k \in [5]$, $\mathcal{S}[\langle k \rangle]$ denotes the set of decryption queries which fall under **Case** $\langle k \rangle$ above.

We now state five lemmas for the five separate cases, the proofs of which we defer to Sect. 5.

Lemma 1. *For $i \in \mathcal{S}[1]$, $\Pr_{\mathcal{O}_0}[\mathsf{badD}[i]] \le \dfrac{2}{N}$.*

Lemma 2. *For* $i \in \mathcal{S}[2]$, $\Pr_{\mathcal{O}_0}[\mathit{badD}[i]] \le \dfrac{64\ell_{MAX}+15}{N}$, *as long as* $2\sigma_T \le N$.

Lemma 3. *For* $i \in \mathcal{S}[3]$, $\Pr_{\mathcal{O}_0}[\mathit{badD}[i]] \le \dfrac{10}{N}$, *as long as* $2\sigma_T \le N$.

Lemma 4. *For* $i \in \mathcal{S}[4]$, $\Pr_{\mathcal{O}_0}[\mathit{badD}[i]] \le \dfrac{2}{N} + \dfrac{32\sigma_T^2}{N^2}$.

Lemma 5. *For* $i \in \mathcal{S}[5]$, $\Pr_{\mathcal{O}_0}[\mathit{badD}[i]] \le \dfrac{64\ell_{MAX}+4}{N} + \dfrac{32\sigma_T^2}{N^2}$.

Taking the maximum over these bounds gives (4), and completes the proof of the claim. □

5 Proof of Lemmas

We recall that for the i-th decryption query, we first sample/set the inputs and outputs of π, and then define $\mathsf{M}_p'^i$ as

$$\mathsf{M'}_p^i := \begin{cases} \mathsf{X'}_p^i + \mathit{\Delta}_p'^i, & p \in [\ell'^i], \\ \mathsf{chop}_{|\mathsf{C'}_p^i|}(\mathsf{Y'}_p^i + \overline{\mathsf{C'}_p^i})||10*, & p = *, i \in \mathcal{I}'. \end{cases}$$

Finally, we set

$$\mathsf{M'}_{\mathsf{tag}}^i := \sum_{p \in \mathcal{P}'^i(-)} \mathsf{M'}_p^i,$$
$$\mathsf{M''}_{\mathsf{tag}}^i := \mathsf{X'}_{\mathsf{tag}}^i + \mathit{\Delta}_{\mathsf{tag}}'^i,$$

and $\mathsf{badD}[i]$ is triggered when $\mathsf{M'}_{\mathsf{tag}}^i = \mathsf{M''}_{\mathsf{tag}}^i$.

The Subcase Tree. In Subsect. 4.5, we divide the set [q'] of decryption queries into five subsets $\mathcal{S}[1], \ldots, \mathcal{S}[5]$, depending on which of five cases a particular decryption query satisfies. We divide each of the cases, except **Case 1**, into various sub-cases. Whenever a $\mathsf{X'}_p^i$ is trivially determined for $p \in [\ell'^i]$, we let i' be such that $\mathsf{X'}_p^i$ is i'-trivial, and whenever $\mathsf{X'}_{\mathsf{tag}}^i$ is trivially determined, we let j be such that $\mathsf{X'}_{\mathsf{tag}}^i$ is j-trivial. (Note that there can be exactly one choice for each of i' and j.)

- **Case 2.** Here $\mathsf{X'}_{p_0}^i = \pi^{-1}(\mathsf{C'}_{p_0}^i + \mathit{\Delta}_{p_0}^{i'})$, so we branch based on $\mathsf{C'}_{p_0}^i + \mathit{\Delta}_{p_0}^{i'}$:
  - *Subcase 2(a).* $\mathsf{C'}_{p_0}^i + \mathit{\Delta}_{p_0}^{i'} \notin \mathsf{Ran}(\pi)$;
  - *Subcase 2(b).* $\mathsf{C'}_{p_0}^i + \mathit{\Delta}_{p_0}^{i'} = \mathsf{KN}^{j'}$ for some $j' \in [q]$;
  - *Subcase 2(c).* $\mathsf{C'}_{p_0}^i + \mathit{\Delta}_{p_0}^{i'} = \mathsf{V}_{s_0}^{j'}$ for some $j' \in [q], s_0 \in [k^{j'}]$;
  - *Subcase 2(d).* $\mathsf{C'}_{p_0}^i + \mathit{\Delta}_{p_0}^{i'} = \mathsf{C}_{p_1}^{j'} + \mathit{\Delta}_{p_1}^{j'}$ for some $j' \in [q], p_1 \in [\ell^{j'}]$;
  - *Subcase 2(e).* $\mathsf{C'}_{p_0}^i + \mathit{\Delta}_{p_0}^{i'} = \mathsf{auth}^{j'} + \mathsf{T}^{j'}$ for some $j' \in [q]$.

- **Case 3.** Here $\mathsf{X'}^i_{p_0} = \mathsf{X'}^i_{\mathsf{tag}} = \pi^{-1}(\mathsf{auth'}^i + \mathsf{T'}^i)$, so we branch based on $\mathsf{auth'}^i + \mathsf{T'}^i$:
  - <u>*Subcase 3(a).*</u> $\mathsf{auth'}^i + \mathsf{T'}^i \notin \mathsf{Ran}(\pi)$;
  - <u>*Subcase 3(b).*</u> $\mathsf{auth'}^i + \mathsf{T'}^i = \mathsf{KN}^{j'}$ for some $j' \in [q]$;
  - <u>*Subcase 3(c).*</u> $\mathsf{auth'}^i + \mathsf{T'}^i = \mathsf{V}^{j'}_{s_0}$ for some $j' \in [q], s_0 \in [k^{j'}]$;
  - <u>*Subcase 3(d).*</u> $\mathsf{auth'}^i + \mathsf{T'}^i = \mathsf{C}^{j'}_{p_1} + \Delta^{j'}_{p_1}$ for some $j' \in [q], p_1 \in [\ell^{j'}]$;
  - <u>*Subcase 3(e).*</u> $\mathsf{auth'}^i + \mathsf{T'}^i = \mathsf{auth}^{j'} + \mathsf{T}^{j'}$ for some $j' \in [q]$.
- **Case 4.** Here $\mathsf{X'}^i_{p_0} = \pi^{-1}(\mathsf{C'}^i_{p_0} + \Delta^{i'}_{p_0})$ and $\mathsf{X'}^i_{\mathsf{tag}} = \pi^{-1}(\mathsf{auth'}^i + \mathsf{T'}^i)$, so we can branch based on $\mathsf{C'}^i_{p_0} + \Delta^{i'}_{p_0}$ and $\mathsf{auth'}^i + \mathsf{T'}^i$ and get seventeen cases here: one covering either of them being randomly sampled, and the other sixteen a Cartesian product between *Subcases 2(b)–2(e)* and *Subcases 3(b)–3(e)*. However, most of this cases can be settled using near-identical arguments, so we make the case division to reflect the interesting cases:
  - <u>*Subcase 4(a).*</u> $\mathsf{C'}^i_{p_0} + \Delta^{i'}_{p_0} \notin \mathsf{Ran}(\pi)$ or $\mathsf{auth'}^i + \mathsf{T'}^i \notin \mathsf{Ran}(\pi)$;
  - <u>*Subcase 4(b).*</u> $\mathsf{C'}^i_{p_0} + \Delta^{i'}_{p_0} \in \mathsf{Ran}(\pi)$, $\mathsf{auth'}^i + \mathsf{T'}^i = \mathsf{V}^{j'}_{s_0}$ for some $j' \in [q], s_0 \in [k^{j'}]$;
  - <u>*Subcase 4(c).*</u> $\mathsf{C'}^i_{p_0} + \Delta^{i'}_{p_0} \in \mathsf{Ran}(\pi)$, $\mathsf{auth'}^i + \mathsf{T'}^i \in \mathsf{Ran}(\pi)$, $\mathsf{auth'}^i + \mathsf{T'}^i \neq \mathsf{V}^{j'}_{s_0}$ for all $j' \in [q], s_0 \in [k^{j'}]$.
- **Case 5.** Here $\mathsf{X'}^i_{p_0} = \pi^{-1}(\mathsf{C'}^i_{p_0} + \Delta^{i'}_{p_0})$ and $\mathsf{X'}^i_{p_1} = \pi^{-1}(\mathsf{C'}^i_{p_1} + \Delta^{i'}_{p_1})$, so we can branch based on $\mathsf{C'}^i_{p_0} + \Delta^{i'}_{p_0}$ and $\mathsf{C'}^i_{p_1} + \Delta^{i'}_{p_1}$:
  - <u>*Subcase 5(a).*</u> $\mathsf{C'}^i_{p_0} + \Delta^{i'}_{p_0} \notin \mathsf{Ran}(\pi)$ or $\mathsf{C'}^i_{p_1} + \Delta^{i'}_{p_1} \notin \mathsf{Ran}(\pi)$;
  - <u>*Subcase 5(b).*</u> $\mathsf{C'}^i_{p_0} + \Delta^{i'}_{p_0} = \mathsf{C}^{j'}_{p_2} + \Delta^{j'}_{p_2}, \mathsf{C'}^i_{p_1} + \Delta^{i'}_{p_1} = \mathsf{C}^{j''}_{p_3} + \Delta^{j''}_{p_3}$ for some $j', j'' \in [q], j' \neq j'', p_2 \in [\ell^{j'}], p_3 \in [\ell^{j''}]$;
  - <u>*Subcase 5(c).*</u> $\mathsf{C'}^i_{p_0} + \Delta^{i'}_{p_0} = \mathsf{C}^{j'}_{p_2} + \Delta^{j'}_{p_2}, \mathsf{C'}^i_{p_1} + \Delta^{i'}_{p_1} = \mathsf{C}^{j'}_{p_3} + \Delta^{j'}_{p_3}$ for some $j' \in [q], p_2, p_3 \in [\ell^{j'}]$;
  - <u>*Subcase 5(d).*</u> $\mathsf{C'}^i_{p_0} + \Delta^{i'}_{p_0} \in \mathsf{Ran}(\pi)$ and $\mathsf{C'}^i_{p_1} + \Delta^{i'}_{p_1} \in \mathsf{Ran}(\pi)$, either $\mathsf{C'}^i_{p_1} + \Delta^{i'}_{p_1} \neq \mathsf{C}^{j'}_{p_2} + \Delta^{j'}_{p_2}$ for any $j' \in [q], p_2 \in [\ell^{j'}]$ or $\mathsf{C'}^i_{p_1} + \Delta^{i'}_{p_1} \neq \mathsf{C}^{j'}_{p_2} + \Delta^{j'}_{p_2}$ for any $j' \in [q], p_2 \in [\ell^{j'}]$.

Now we turn to the proof of the lemmas. We make the following observations:

- When $\mathsf{X'}^i_p$ is i'-trivial for some $p \in [\ell'^i]$, $\mathsf{M'}^i_p = \mathsf{M}^{i'}_p$;
- When $\mathsf{X'}^i_{\mathsf{tag}}$ is j-trivial, $\mathsf{M'}^i_{\mathsf{tag}} + \Delta'^i_{\mathsf{tag}} = \mathsf{M}^j_{\mathsf{tag}} + \Delta^j_{\mathsf{tag}}$.

For brevity, when we write the collision equation(s) that need to be satisfied for $\mathsf{badD}[i]$ to occur, the random variables that contribute to the subsequent probability calculation are indicated $\underbrace{\text{thus}}$.

5.1 Proof of Lemma 1

Lemma 1. *For* $i \in \mathcal{S}[1]$, $\Pr_{\mathcal{O}_0}[\mathit{badD}[i]] \leq \dfrac{2}{N}$.

Proof. When $i \in \mathcal{S}[1]$, $\mathsf{X'}^{i}_{p}$ is trivially determined for all $p \in \mathcal{P}'^{i}$. From observations above, for $\mathsf{badD}[i]$ to occur, we must have

$$\sum_{p\in[\ell^{i'}]} \mathsf{M}^{i'}_{p} + \Delta'^{i}_{\mathsf{tag}} = \mathsf{M}^{j}_{\mathsf{tag}} + \Delta^{j}_{\mathsf{tag}},$$

or $$\mathsf{Q}^{i'} + \mathsf{Q}^{j} + (\lambda^{\$}_{\ell'^{i}} + \lambda^{\$}_{\ell^{j}}) \cdot \mathsf{L} = \sum_{p\in[\ell^{i'}]} \mathsf{M}^{i'}_{p} + \mathsf{M}^{j}_{\mathsf{tag}}.$$

If $\ell^{j} \neq \ell'^{i}$, the equation is

$$\mathbf{H}^{i'} \cdot \mathsf{KN}^{i'} + \mathbf{H}^{j} \cdot \mathsf{KN}^{j} + (\lambda^{\$}_{\ell'^{i}} + \lambda^{\$}_{\ell^{j}}) \cdot \underbrace{\mathsf{L}} = \sum_{p\in[\ell^{i'}]} \mathsf{M}^{i'}_{p} + \mathsf{M}^{j}_{\mathsf{tag}},$$

where the coefficient of L is non-zero. The probability of this $\leq 1/(N-2) \leq 1/2N$. If $\ell^{j} = \ell'^{i}$, so $j \neq i'$ (the decryption query has to be non-trivial), but $\mathsf{TN}^{j} = \mathsf{TN}^{i'}$, then $\mathbf{H}^{j} + \mathbf{H}^{i'}$ is full-rank, so

$$(\mathbf{H}^{i'} + \mathbf{H}^{j}) \cdot \underbrace{\mathsf{KN}^{j}} = \sum_{p\in[\ell^{i'}]} \mathsf{M}^{i'}_{p} + \mathsf{M}^{j}_{\mathsf{tag}}.$$

The probability of this $\leq 1/N$. And finally when $\ell^{j} = \ell'^{i}$ and $\mathsf{TN}^{j} \neq \mathsf{TN}^{i'}$, we have

$$\mathbf{H}^{i'} \cdot \underbrace{\mathsf{KN}^{i'}} + \mathbf{H}^{j} \cdot \mathsf{KN}^{j} = \sum_{p\in[\ell^{i'}]} \mathsf{M}^{i'}_{p} + \mathsf{M}^{j}_{\mathsf{tag}}.$$

The probability of this $\leq 1/(N-1) \leq 2/N$. This completes the proof. □

5.2 Proof of Lemma 2

Lemma 2. *For* $i \in \mathcal{S}[2]$, $\Pr_{\mathcal{O}_0}[\mathit{badD}[i]] \leq \dfrac{64\ell_{MAX}+15}{N}$, *as long as* $2\sigma_T \leq N$.

Proof. When $i \in \mathcal{S}[2]$, for some $p_0 \in [\ell'^{i}]$, $\mathsf{X'}^{i}_{p}$ is trivially determined for all $p \in \mathcal{P}'^{i} \setminus \{p_0\}$, but $\mathsf{X'}^{i}_{p_0}$ is not trivially determined . The equation for $\mathsf{badD}[i]$ becomes

$$\sum_{p\in[\ell^{i'}]\setminus\{p_0\}} \mathsf{M}^{i'}_{p} + \pi^{-1}(\mathsf{C'}^{i}_{p_0} + \Delta^{i'}_{p_0}) + \Delta^{i'}_{p_0} + \Delta^{i'}_{\mathsf{tag}} = \mathsf{M}^{j}_{\mathsf{tag}} + \Delta^{j}_{\mathsf{tag}}$$

or $$\pi^{-1}(\mathsf{C'}^{i}_{p_0} + \Delta^{i'}_{p_0}) + \mathsf{Q}^{j} + (\lambda_{p_0} + \lambda^{\$}_{\ell^{i'}} + \lambda^{\$}_{\ell^{j}}) \cdot \mathsf{L} = \mathsf{MM},$$

where

$$\mathsf{MM} := \sum_{p\in[\ell^{i'}]\setminus\{p_0\}} \mathsf{M}^{i'}_{p} + \mathsf{M}^{j}_{\mathsf{tag}}.$$

Based on the value of $\mathsf{C'}^{i}_{p_0} + \Delta^{i'}_{p_0}$, we look at the subcases listed in the tree at the beginning of this section. Note that

$$\Delta^{i'}_{p_0} + \Delta^{i'}_{\mathsf{tag}} + \Delta^{j}_{\mathsf{tag}} = \mathsf{Q}^{j} + (\lambda_{p_0} + \lambda^{\$}_{\ell^{i'}} + \lambda^{\$}_{\ell^{j}}) \cdot \mathsf{L}.$$

- *Subcase 2(a).* $\mathsf{C'}^{i}_{p_0} + \mathit{\Delta}^{i'}_{p_0} \notin \mathsf{Ran}(\pi)$, so $\mathsf{X'}^{i}_{p_0}$ is sampled. The equation for $\mathsf{badD}[i]$ is
$$\underbrace{\mathsf{X'}^{i}_{p_0}} + \mathbf{H}^{j} \cdot \mathsf{KN}^{j} + (\lambda_{p_0} + \lambda^{\$}_{\ell^{i'}} + \lambda^{\$}_{\ell^{j}}) \cdot \mathsf{L} = \mathsf{MM}.$$
The probability of this $\leq 1/(N-2) \leq 2/N$.

- *Subcase 2(b).* $\mathsf{C'}^{i}_{p_0} + \mathit{\Delta}^{i'}_{p_0} = \mathsf{KN}^{j'}$ for some $j' \in [q]$, so $\mathsf{X'}^{i}_{p_0} = \mathsf{TN}^{j'}$, and a second equation comes from the condition for $\mathsf{badD}[i]$. When $\ell^{i'} = \ell^{j}$ and $\mathsf{TN}^{i'} = \mathsf{TN}^{j'}$, these two equations become
$$(\mathbf{I} + \mathbf{H}^{i'}) \cdot \underbrace{\mathsf{KN}^{j'}} + \lambda_{p_0} \cdot \underbrace{\mathsf{L}} = \mathsf{C'}^{i}_{p_0},$$
$$\mathbf{H}^{j} \cdot \mathsf{KN}^{j} + \lambda_{p_0} \cdot \underbrace{\mathsf{L}} = \mathsf{TN}^{j'} + \mathsf{MM}.$$
Since there are at most 2^6 choices for j', this probability $\leq 64/(N-1)(N-2) \leq 256/N^2 \leq 1/N$. When $\ell^{i'} = \ell^{j}$ and $\mathsf{TN}^{i'} \neq \mathsf{TN}^{j'}$, the two equations become
$$\underbrace{\mathsf{KN}^{j'}} + \mathbf{H}^{i'} \cdot \mathsf{KN}^{i'} + \lambda_{p_0} \cdot \mathsf{L} = \mathsf{C'}^{i}_{p_0},$$
$$\mathbf{H}^{j} \cdot \mathsf{KN}^{j} + \lambda_{p_0} \cdot \mathsf{L} = \mathsf{TN}^{j'} + \mathsf{MM}.$$
Here there are q choices for j', and this probability $\leq q(N-2)(N-3) \leq 4q/N^2$. When $\ell^{i'} \neq \ell^{j}$, the two equations become
$$\underbrace{\mathsf{KN}^{j'}} + \mathbf{H}^{i'} \cdot \mathsf{KN}^{i'} + \lambda_{p_0} \cdot \underbrace{\mathsf{L}} = \mathsf{C'}^{i}_{p_0},$$
$$\mathbf{H}^{j} \cdot \mathsf{KN}^{j} + (\lambda_{p_0} + \lambda^{\$}_{\ell^{i'}} + \lambda^{\$}_{\ell^{j}}) \cdot \underbrace{\mathsf{L}} = \mathsf{TN}^{j'} + \mathsf{MM}.$$
Here too there are q choices for j', and this probability $\leq q/(N-2)(N-3) \leq 4q/N^2$. Thus, the probability of badB and *Subcase 2(b)* simultaneously happening is at most $2/N$, as long as $2q \leq N$.

- *Subcase 2(c).* $\mathsf{C'}^{i}_{p_0} + \mathit{\Delta}^{i'}_{p_0} = \mathsf{V}^{j'}_{s_0}$ for some $j' \in [q], s_0 \in [k^{j'}]$, so $\mathsf{X'}^{i}_{p_0} = \mathsf{U}^{j'}_{s_0}$, and a second equation comes from the condition for $\mathsf{badD}[i]$. Here the two equations are
$$\underbrace{\mathsf{V}^{j'}_{s_0}} + \mathbf{H}^{i'} \cdot \mathsf{KN}^{i'} + \lambda_{p_0} \cdot \mathsf{L} = \mathsf{C'}^{i}_{p_0},$$
$$\mathbf{H}^{j} \cdot \underbrace{\mathsf{KN}^{j}} + (\lambda_{s_0} + \lambda_{p_0} + \lambda^{\$}_{\ell^{i'}} + \lambda^{\$}_{\ell^{j}}) \cdot \mathsf{L} = \mathsf{A}^{j'}_{s_0} + \mathsf{MM}.$$
There are α choices for (j', s_0), so the probability of this $\leq \alpha/(N-2)(N-3) \leq 4\alpha/N^2 \leq 2/N$, as long as $2\alpha \leq N$.

– *Subcase 2(d).* $\mathsf{C}'^{i}_{p_0} + \Delta^{i'}_{p_0} = \mathsf{C}^{j'}_{p_1} + \Delta^{j'}_{p_1}$ for some $j' \in [q], p_1 \in [\ell^{j'}]$, so $\mathsf{X}'^{i}_{p_0} = \mathsf{X}^{j'}_{p_1}$, and a second equation comes from the condition for $\mathsf{badD}[i]$. The two equations are

$$\mathsf{Q}^{i'} + \mathsf{Q}^{j'} + (\lambda_{p_1} + \lambda_{p_0}) \cdot \mathsf{L} = \mathsf{C}^{j'}_{p_1} + \mathsf{C}'^{i}_{p_0},$$
$$\mathsf{Q}^{j} + \mathsf{Q}^{j'} + (\lambda_{p_1} + \lambda_{p_0} + \lambda^{\$}_{\ell^{i'}} + \lambda^{\$}_{\ell^{j}}) \cdot \mathsf{L} = \mathsf{M}^{j'}_{p_1} + \mathsf{MM}.$$

When $i' = j$, the two equations become

$$\mathbf{H}^{i'} \cdot \mathsf{KN}^{i'} + \mathbf{H}^{j'} \cdot \mathsf{KN}^{j'} + (\lambda_{p_1} + \lambda_{p_0}) \cdot \underbrace{\mathsf{L}} = \mathsf{C}^{j'}_{p_1} + \mathsf{C}'^{i}_{p_0},$$
$$\mathbf{H}^{i'} \cdot \mathsf{KN}^{i'} + \mathbf{H}^{j'} \cdot \mathsf{KN}^{j'} + (\lambda_{p_1} + \lambda_{p_0}) \cdot \underbrace{\mathsf{L}} = \mathsf{M}^{j'}_{p_1} + \mathsf{MM},$$

which is actually a single equation with the constraint that $\mathsf{C}^{j'}_{p_1} + \mathsf{C}'^{i}_{p_0} = \mathsf{M}^{j'}_{p_1} + \mathsf{MM}$, which implies that j' must satisfy $\mathsf{M}^{j'}_{p_1} + \mathsf{C}^{j'}_{p_1} = \mathsf{C}'^{i}_{p_0} + \mathsf{MM}$. Since there are no collisions on $\mathsf{M}^{j'}_{p_1} + \mathsf{C}^{j'}_{p_1}$ over all pairs (j', p_1), there is at most one choice of (j', p_1) satisfying this. Thus, the probability of this $\leq 1/(N-2) \leq 2/N$. When $i \neq j'$ but $\mathsf{KN}^{i'} = \mathsf{KN}^{j}$ and $\ell^{i'} = \ell^{j}$, the two equations become

$$\mathbf{H}^{i'} \cdot \underbrace{\mathsf{KN}^{i'}} + \mathbf{H}^{j'} \cdot \mathsf{KN}^{j'} + (\lambda_{p_1} + \lambda_{p_0}) \cdot \underbrace{\mathsf{L}} = \mathsf{C}^{j'}_{p_1} + \mathsf{C}'^{i}_{p_0},$$
$$\mathbf{H}^{j} \cdot \underbrace{\mathsf{KN}^{i'}} + \mathbf{H}^{j'} \cdot \mathsf{KN}^{j'} + (\lambda_{p_1} + \lambda_{p_0}) \cdot \underbrace{\mathsf{L}} = \mathsf{M}^{j'}_{p_1} + \mathsf{MM}.$$

There are at most σ choices for (j', p_1), so the probability $\leq \sigma/(N-1)(N-2) \leq 4\sigma/N^2$. When $i \neq j'$, $\mathsf{KN}^{i'} = \mathsf{KN}^{j} \neq \mathsf{KN}^{j'}$, $\ell^{i'} \neq \ell^{j}$, the two equations become

$$\mathbf{H}^{i'} \cdot \underbrace{\mathsf{KN}^{i'}} + \mathbf{H}^{j'} \cdot \mathsf{KN}^{j'} + (\lambda_{p_1} + \lambda_{p_0}) \cdot \underbrace{\mathsf{L}} = \mathsf{C}^{j'}_{p_1} + \mathsf{C}'^{i}_{p_0},$$
$$\mathbf{H}^{j} \cdot \underbrace{\mathsf{KN}^{i'}} + \mathbf{H}^{j'} \cdot \mathsf{KN}^{j'} + (\lambda_{p_1} + \lambda_{p_0} + \lambda^{\$}_{\ell^{i'}} + \lambda^{\$}_{\ell^{j}}) \cdot \underbrace{\mathsf{L}} = \mathsf{M}^{j'}_{p_1} + \mathsf{MM}.$$

There are at most σ choices for (j', p_1), so the probability of this $\leq \sigma/(N-1)(N-2) \leq 4\sigma/N^2$. When $i \neq j'$, $\mathsf{KN}^{i'} = \mathsf{KN}^{j} = \mathsf{KN}^{j'}$, $\ell^{i'} \neq \ell^{j}$, the two equations become

$$(\mathbf{H}^{i'} + \mathbf{H}^{j'}) \cdot \underbrace{\mathsf{KN}^{i'}} + (\lambda_{p_1} + \lambda_{p_0}) \cdot \underbrace{\mathsf{L}} = \mathsf{C}^{j'}_{p_1} + \mathsf{C}'^{i}_{p_0},$$
$$(\mathbf{H}^{j} + \mathbf{H}^{j'}) \cdot \underbrace{\mathsf{KN}^{i'}} + (\lambda_{p_1} + \lambda_{p_0} + \lambda^{\$}_{\ell^{i'}} + \lambda^{\$}_{\ell^{j}}) \cdot \underbrace{\mathsf{L}} = \mathsf{M}^{j'}_{p_1} + \mathsf{MM}.$$

They may be multiples of the same equation, in which case we have at most 64 choices for j' and at most ℓ_{MAX} choices for p_1. Then this probability $\leq 64\ell_{\mathrm{MAX}}/N$. When they are different equations, this probability $\leq \sigma/N(N-1) \leq 2\sigma/N^2$. So the probability of $\mathsf{badD}[i]$ and *Subcase 2(d)* simultaneously happening is at most $(64\ell_{\mathrm{MAX}} + 7)/N$ as long as $2\sigma \leq N$.

- *Subcase 2(e).* $\mathsf{C'}^i_{p_0} + \Delta^{i'}_{p_0} = \mathsf{auth}^{j'} + \mathsf{T}^{j'}$ for some $j' \in [q]$, so $\mathsf{X'}^i_{p_0} = \mathsf{X}^{j'}_{\mathsf{tag}}$, and a second equation comes from the condition for $\mathsf{badD}[i]$. When $j = j'$, the two equations become

$$\underbrace{\mathsf{V}^j_1} + \sum_{s\in[2..k^j]} \mathsf{V}^j_s + \mathbf{H}^{i'} \cdot \mathsf{KN}^{i'} + \lambda_{p_0} \cdot \underbrace{\mathsf{L}} = \mathsf{T}^{j'} + \mathsf{C'}^i_{p_0},$$

$$(\lambda_{p_0} + \lambda^{\$}_{\ell^{i'}}) \cdot \underbrace{\mathsf{L}} = \mathsf{M}^{j'}_{\mathsf{tag}} + \mathsf{MM},$$

and the probability of this $\leq 1/(N - k^j - 1)(N - k^j - 2) \leq 4/N^2$. When $j \neq j'$, the two equations become

$$\underbrace{\mathsf{V}^{j'}_1} + \sum_{s\in[2..k^{j'}]} \mathsf{V}^{j'}_s + \mathbf{H}^{i'} \cdot \mathsf{KN}^{i'} + \lambda_{p_0} \cdot \mathsf{L} = \mathsf{T}^{j'} + \mathsf{C'}^i_{p_0},$$

$$\mathbf{H}^j \cdot \mathsf{KN}^j + \mathbf{H}^{j'} \cdot \underbrace{\mathsf{KN}^{j'}} + (\lambda^{\$}_{\ell^{j'}} + \lambda_{p_0} + \lambda^{\$}_{\ell^{i'}} + \lambda^{\$}_{\ell^j}) \cdot \mathsf{L} = \mathsf{M}^{j'}_{\mathsf{tag}} + \mathsf{MM}.$$

and the probability of this $\leq q/(N - k^{j'} - 3)(N - k^{j'} - 4) \leq 4q/N^2$. Thus, the probability of badB and *Subcase 2(e)* simultaneously happening is at most $2/N$, as long as $2q \leq N$.

Summing over the subcases completes the proof. □

5.3 Proof of Lemma 3

Lemma 3. *For $i \in \mathcal{S}[3]$, $\Pr_{\mathcal{O}_0}[\mathsf{badD}[i]] \leq \dfrac{10}{N}$, as long as $2\sigma_T \leq N$.*

Proof. When $i \in \mathcal{S}[3]$, $\mathsf{X'}^i_p$ is trivially determined for all $p \in [\ell'^i]$, but $\mathsf{X'}^i_{\mathsf{tag}}$ is not trivially determined. The equation for $\mathsf{badD}[i]$ becomes

$$\pi^{-1}(\mathsf{auth'}^i + \mathsf{T'}^i) + \Delta^{i'}_{\mathsf{tag}} = \sum_{p\in[\ell^{i'}]} \mathsf{M}^{i'}_p.$$

Based on the value of $\mathsf{auth'}^i + \mathsf{T'}^i$, we look at the subcases listed in the tree at the beginning of this section.

- *Subcase 3(a).* $\mathsf{auth'}^i + \mathsf{T'}^i \notin \mathsf{Ran}(\pi)$, so $\mathsf{X'}^i_{\mathsf{tag}}$ is sampled. The equation for $\mathsf{badD}[i]$ is

$$\underbrace{\mathsf{X'}^i_{\mathsf{tag}}} + \mathbf{H}^{i'} \cdot \mathsf{KN}^{i'} + \lambda^{\$}_{\ell^{i'}} \cdot \mathsf{L} = \sum_{p\in[\ell^{i'}]} \mathsf{M}^{i'}_p.$$

The probability of this $\leq 1/(N - 2) \leq 2/N$.

- *Subcase 3(b).* $\mathsf{auth'}^i + \mathsf{T'}^i = \mathsf{KN}^{j'}$ for some $j' \in [q]$, so $\mathsf{X'}^i_{\mathsf{tag}} = \mathsf{TN}^{j'}$, and a second equation comes from the condition for $\mathsf{badD}[i]$. The two equations are

$$\underbrace{\mathsf{KN}^{j'}} + \sum_{s \in [k'^i]} \mathsf{V'}^i_s = \mathsf{T'}^i,$$

$$\mathbf{H}^{i'} \cdot \mathsf{KN}^{i'} + \lambda^{\$}_{\ell^{i'}} \cdot \underbrace{\mathsf{L}} = \mathsf{TN}^{j'} + \sum_{p \in [\ell^{i'}]} \mathsf{M}^{i'}_p.$$

  The probability of this $\leq 4q/N^2 \leq 2/N$, as long as $2q \leq N$.

- *Subcase 3(c).* $\mathsf{auth'}^i + \mathsf{T'}^i = \mathsf{V}^{j'}_{s_0}$ for some $j' \in [q], s_0 \in [k^{j'}]$, so $\mathsf{X'}^i_{p_0} = \mathsf{U}^{j'}_{s_0}$, and a second equation comes from the condition for $\mathsf{badD}[i]$. The two equations are

$$\underbrace{\mathsf{V}^{j'}_{s_0}} + \sum_{s \in [k'^i]} \mathsf{V'}^i_s = \mathsf{T'}^i,$$

$$\mathbf{H}^{i'} \cdot \mathsf{KN}^{i'} + (\lambda_{s_0} + \lambda^{\$}_{\ell^{i'}}) \cdot \underbrace{\mathsf{L}} = \mathsf{A}^{j'}_{s_0} + \sum_{p \in [\ell^{i'}]} \mathsf{M}^{i'}_p.$$

  When the top equation vanishes, the probability of this $\leq 1/(N-1) \leq 2/N$. When both equations are there, the probability of this $\leq 4\alpha/N^2$. So the probability of $\mathsf{badD}[i]$ and *Subcase 3(c)* simultaneously happening is at most $2/N$, as long as $2\alpha \leq N$.

- *Subcase 3(d).* $\mathsf{auth'}^i + \mathsf{T'}^i = \mathsf{C}^{j'}_{p_1} + \Delta^{j'}_{p_1}$ for some $j' \in [q], p_1 \in [\ell^{j'}]$, so $\mathsf{X'}^i_{\mathsf{tag}} = \mathsf{X}^{j'}_{p_1}$, and a second equation comes from the condition for $\mathsf{badD}[i]$. The two equations are

$$\underbrace{\mathsf{V'}^i_1} + \sum_{s \in [2..k'^i]} \mathsf{V'}^i_s + \mathbf{H}^{j'} \cdot \mathsf{KN}^{j'} + \lambda_{p_1} \cdot \mathsf{L} = \mathsf{C}^{j'}_{p_1} + \mathsf{T'}^i,$$

$$\mathbf{H}^{j'} \cdot \mathsf{KN}^{j'} + \mathbf{H}^{i'} \cdot \mathsf{KN}^{i'} + (\lambda_{p_1} + \lambda^{\$}_{\ell^{i'}}) \cdot \underbrace{\mathsf{L}} = \mathsf{M}^{j'}_{p_1} + \sum_{p \in [\ell^{i'}]} \mathsf{M}^{i'}_p.$$

  The probability of this $\leq 4\sigma/N^2 \leq 2/N$, as long as $2\sigma \leq N$.

- *Subcase 3(e).* $\mathsf{auth'}^i + \mathsf{T'}^i = \mathsf{auth}^{j'} + \mathsf{T}^{j'}$ for some $j' \in [q]$, so $\mathsf{X'}^i_{\mathsf{tag}} = \mathsf{X}^{j'}_{\mathsf{tag}}$, and a second equation comes from the condition for $\mathsf{badD}[i]$. If $\mathsf{A}^{j'} = \mathsf{A'}^i$, we have $\mathsf{T}^{j'} \neq \mathsf{T'}^i$ (by definition of this case), or $\mathsf{auth}^{j'} \neq \mathsf{auth'}^i$, a contradiction. So $\mathsf{A}^{j'} \neq \mathsf{A'}^i$. If we can find $s_0 \leq k^{j'}$ such that either $s_0 > k'^i$, or $\mathsf{A}^{j'}_{s_0} \neq \mathsf{A'}^i_{s_0}$, then the equations are

$$\underbrace{\mathsf{V}^{j'}_{s_0}} + \sum_{s \in [k'^i]} \mathsf{V'}^i_s + \sum_{s \in [k^{j'}] \backslash \{s_0\}} \mathsf{V}^{j'}_s = \mathsf{T'}^i + \mathsf{T}^{j'},$$

$$\mathbf{H}^{j'} \cdot \mathsf{KN}^{j'} + \mathbf{H}^{i'} \cdot \mathsf{KN}^{i'} + (\lambda^{\$}_{\ell^{j'}} + \lambda^{\$}_{\ell^{i'}}) \cdot \underbrace{\mathsf{L}} = \mathsf{M}^{j'}_{\mathsf{tag}} + \sum_{p \in [\ell^{i'}]} \mathsf{M}^{i'}_p.$$

Otherwise we can find $s_0 \leq k'^i$ such that either $s_0 > k^{j'}$, or $\mathsf{A'}^i_{s_0} \neq \mathsf{A}^{j'}_{s_0}$, then the equations are

$$\underbrace{\mathsf{V'}^i_{s_0}} + \sum_{s \in [k'^i] \setminus \{s_0\}} \mathsf{V'}^i_s + \sum_{s \in [k^{j'}]} \mathsf{V}^{j'}_s = \mathsf{T'}^i + \mathsf{T}^{j'},$$

$$\mathbf{H}^{j'} \cdot \mathsf{KN}^{j'} + \mathbf{H}^{i'} \cdot \mathsf{KN}^{i'} + (\lambda^{\$}_{\ell j'} + \lambda^{\$}_{\ell i'}) \cdot \underbrace{\mathsf{L}} = \mathsf{M}^{j'}_{\mathsf{tag}} + \sum_{p \in [\ell^{i'}]} \mathsf{M}^{i'}_p .$$

The probability of either of these does not exceed $2/N$ as long as $2\sigma \leq N$.

Summing over the subcases completes the proof. □

5.4 Proof of Lemma 4

Lemma 4. *For $i \in \mathcal{S}[4]$,* $\Pr_{\mathcal{O}_0}[\mathsf{badD}[i]] \leq \dfrac{2}{N} + \dfrac{32\sigma_T^2}{N^2}$.

Proof. The bounds for **Case 4** are simple to derive:

- *Subcase 4(a).* We get a bound of $2/N$ on the probability as in *Subcase 2(a)* or *Subcase 3(a)*.

- *Subcase 4(b).* We treat this separately because the equation

  $$\mathsf{auth'}^i + \mathsf{T'}^i = \mathsf{V}^{j'}_{s_0}$$

  can vanish. When it does not vanish, we can find two independent collision equations that need to be satisfied, which can occur together with a probability of at most $2/N^2$. When it does vanish, we proceed as in *Subcase 3(c)* and use instead the equation where $\mathsf{Q}^{i'} + (\lambda_{s_0} + \lambda^{\$}_{\ell i'}) \cdot \mathsf{L}$ is equated to a constant. There are four possibilities here based on *Subcases 2(b)-2(e)*. In each possibility, there are at most σ_T^2 choices for these collision indices. Thus, each possibility with $\mathsf{badD}[i]$ has a probability of at most $2\sigma_T^2/N^2$, and *Subcase 4(b)* has a probability of at most $8\sigma_T^2/N^2$.

- *Subcase 4(c).* There are twelve possibilites here, based on various combinations of *Subcases 2(b)-2(e)* and *Subcases 3(b),3(d),3(e)*. In each of these possibilities, we can find two independent collision equations that need to be satisfied, which can occur together with a probability of at most $2/N^2$. There are at most σ_T^2 choices for these collision indices. Thus, each possibility with $\mathsf{badD}[i]$ has a probability of at most $2\sigma_T^2/N^2$, and *Subcase 4(c)* has a probability of at most $24\sigma_T^2/N^2$.

Summing over the three subcases completes the proof. □

5.5 Proof of Lemma 5

Lemma 5. *For $i \in \mathcal{S}[5]$,* $\Pr_{\mathcal{O}_0}[\mathsf{badD}[i]] \leq \dfrac{64\ell_{MAX} + 4}{N} + \dfrac{32\sigma_T^2}{N^2}$.

Proof. We look one by one at the subcases listed in the tree at the beginning of this section.

- *Subcase 5(a).* Here we get a bound of $2/N$ on the probability as in *Subcase 2(a)*.
- *Subcase 5(b).* This is the case when

$$\mathsf{C'}^{i}_{p_0} + \mathsf{Q}^{i'} + \lambda_{p_0} \cdot \mathsf{L} = \mathsf{C}^{j'}_{p_2} + \mathsf{Q}^{j'} + \lambda_{p_2} \cdot \mathsf{L},$$
$$\mathsf{C'}^{i}_{p_1} + \mathsf{Q}^{i'} + \lambda_{p_1} \cdot \mathsf{L} = \mathsf{C}^{j''}_{p_3} + \mathsf{Q}^{j''} + \lambda_{p_3} \cdot \mathsf{L}$$

  for some $j', j'' \in [q], p_2 \in [\ell^{j'}], p_3 \in [\ell^{j''}]$ with $j' \neq j''$. If $\mathsf{TN}^{j'} = \mathsf{TN}^{i'}$, the equations may become dependent on each other. But here there are at most 64 choices for j', since the nonce is distinct in every encryption query, and only 64 distinct values of $\mathsf{N}^{j'}$ can yield the same $\mathsf{TN}^{j'}$. Thus the probability of this does not exceed $64\ell_{\mathrm{MAX}}/N$. Otherwise, we always get two independent equations, and the bound of $2\sigma_T^2/N^2$ holds. Thus, the probability of *Subcase 5(b)* with $\mathsf{badD}[i]$ does not exceed $64\ell_{\mathrm{MAX}}/N + 2\sigma_T^2/N^2$.
- *Subcase 5(c).* This is trickier. Here too these two equations may become the same equation. Since the equations can be rewritten as

$$\mathsf{Q}^{i'} + \mathsf{Q}^{j'} + (\lambda_{p_0} + \lambda_{p_2}) \cdot \mathsf{L} = \mathsf{C'}^{i}_{p_0} + \mathsf{C}^{j'}_{p_2},$$
$$\mathsf{Q}^{i'} + \mathsf{Q}^{j'} + (\lambda_{p_1} + \lambda_{p_3}) \cdot \mathsf{L} = \mathsf{C'}^{i}_{p_1} + \mathsf{C}^{j'}_{p_3},$$

  they become the same equation when $\lambda_{p_0} + \lambda_{p_2} = \lambda_{p_1} + \lambda_{p_3}$ and $\mathsf{C'}^{i}_{p_0} + \mathsf{C'}^{i}_{p_2} = \mathsf{C'}^{i}_{p_1} + \mathsf{C'}^{i}_{p_3}$. Thus any valid choice of (p_2, p_3) must satisfy

$$\lambda_{p_2} + \lambda_{p_3} = \lambda_{p_0} + \lambda_{p_1},$$
$$\mathsf{C'}^{i}_{p_2} + \mathsf{C'}^{i}_{p_3} = \mathsf{C'}^{i}_{p_0} + \mathsf{C'}^{i}_{p_1},$$

  i.e., for each such choice of (p_2, p_3), $\lambda_{p_2} + \lambda_{p_3}$ takes the same fixed value, and $\mathsf{C'}^{i}_{p_2} + \mathsf{C'}^{i}_{p_3}$ take the same fixed value. Since badA has not occurred, we know there are at most 2 such choices of (p_2, p_3). Thus the probability of this does not exceed $2/N$.
- *Subcase 5(d).* There can be fifteen possibilities, depending on various combinations of the subcases of **Case 2**. In each of these, we can find two independent collision equations that need to be satisfied, which can occur together with a probability of at most $2/N^2$. There are at most σ_T^2 choices for these collision indices. Thus, each of those possibilities with $\mathsf{badD}[i]$ has a probability of at most $2\sigma_T^2/N^2$, and *Subcase 5(d)* has a probability of at most $30\sigma_T^2/N^2$.

Summing over the four subcases completes the proof. □

More detailed proofs of Lemmas 4 and 5 can be found in the full version of the paper at the IACR eprint archive, at the url https://eprint.iacr.org/2017/845.pdf.

References

[ABL+13] Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E., Yasuda, K.: Parallelizable and authenticated online ciphers. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8269, pp. 424–443. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42033-7_22

[BGM04] Bellare, M., Goldreich, O., Mityagin, A.: The power of verification queries in message authentication and authenticated encryption. Cryptology ePrint Archive, Report 2004/309 (2004). http://eprint.iacr.org/2004/309

[cae] Caesar competition (2013). https://competitions.cr.yp.to/caesar.html

[DN14] Datta, N., Nandi, M.: ELmE: A misuse resistant parallel authenticated encryption. In: Susilo, W., Mu, Y. (eds.) ACISP 2014. LNCS, vol. 8544, pp. 306–321. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08344-5_20

[GPR17] Gaži, P., Pietrzak, K., Rybár, M.: The exact security of PMAC. IACR Trans. Symmetric Cryptol. **2016**(2), 145–161 (2017)

[HKR15] Hoang, V.T., Krovetz, T., Rogaway, P.: Robust authenticated-encryption AEZ and the problem that it solves. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 15–44. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_2

[KR11] Krovetz, T., Rogaway, P.: The software performance of authenticated-encryption modes. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 306–327. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21702-9_18

[McG08] McGrew, D.: An interface and algorithms for authenticated encryption (2008). https://buildbot.tools.ietf.org/html/rfc5116

[MV04] McGrew, D., Viega, J.: The security and performance of the galois/counter mode (GCM) of operation. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 343–355. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30556-9_27

[MV05] McGrew, D., Viega, J.: The galois/counter mode of operation (gcm). In: NIST Modes Operation Symmetric Key Block Ciphers (2005)

[Pat08] Patarin, J.: The “Coefficients H” technique. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 328–345. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04159-4_21

[RBB03] Rogaway, P., Bellare, M., Black, J.: OCB: A block-cipher mode of operation for efficient authenticated encryption. ACM Trans. Inf. Syst. Secur. **6**(3), 365–403 (2003)

[Rog02] Rogaway, P.: Authenticated-encryption with associated-data. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, NY, USA, pp. 98–107. ACM, New York (2002)

[Rog04] Rogaway, P.: Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 16–31. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30539-2_2

[RS06] Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_23

[WHF02] Whiting, D., Housley, R., Ferguson, N.: AES encryption & authentication using CTR mode & CBC-MAC. IEEE P802, 11 (2002)