Damaging, simplifying, and salvaging p-OMD

Damaging, Simplifying, and Salvaging p-OMD

Tomer Ashur and Bart Mennink

Dept. Electrical Engineering, ESAT/COSIC, KU Leuven, and iMinds, Belgium
{tomer.ashur,bart.mennink}@esat.kuleuven.be

Abstract. One of the submissions to the CAESAR competition for the design of a new authenticated encryption scheme is Offset Merkle-Damgård (OMD). At FSE 2015, Reyhanitabar et al. introduced p-OMD, an improvement of OMD that processes the associated data almost for free. As an extra benefit, p-OMD was claimed to offer integrity against nonce-misusing adversaries, a property that OMD does not have. In this work we show how a nonce-misusing adversary can forge a message for the original p-OMD using only 3 queries (including the forgery). As a second contribution, we generalize and simplify p-OMD. This is done via the introduction of the authenticated encryption scheme Spoed. The most important difference is the usage of a generalized padding function GPAD, which neatly eliminates the need for a case distinction in the design specification and therewith allows for a significantly shorter description of the scheme and a better security bound. Finally, we introduce the authenticated encryption scheme Spoednic, a variant of Spoed providing authenticity against a nonce-misusing adversary at a modest price.

Keywords: Authenticated encryption, CAESAR, p-OMD, nonce-misuse, forgery, simplification

1 Introduction

The principle of authenticated encryption, where both the confidentiality as well as the integrity of data is guaranteed has gained renewed attention in the last couple of years. Emerged from this is the CAESAR competition for the design of new authenticated encryption schemes [5]. CAESAR has received 57 submissions, 30 of which have recently advanced to the second round. Many of these designs have already received further attention via attacks, supporting security proofs, or generalizations.

One of the second round candidates of the CAESAR competition is Offset Merkle-Damgård (OMD) by Cogliani et al. [8,9]. It is characterized by the usage of a full-fledged compression function, and in fact the CAESAR submission takes the SHA256 compression function. OMD is proven to achieve birthday-bound security on the state against adversaries that are not allowed to re-use the nonce. At ProvSec 2014, Reyhanitabar et al. [17] showed how to generalize the scheme to achieve security against nonce-misusing adversaries. On the downside, these schemes are not online and are less efficient than OMD. At FSE 2015 Reyhanitabar et al. [18] presented p-OMD (pure OMD). p-OMD improves over classical OMD in that the associated data is processed almost for free. This is achieved by processing the message blocks as normal message inputs to the compression function and by XORing the associated data into the state. The authors prove that p-OMD inherits all security features of OMD, particularly birthday-bound security against nonce-respecting adversaries. In [19], an early version of [18], it was suggested that p-OMD also offers integrity against nonce-misusing adversaries.

1.1 Nonce-Misuse Forgery on p-OMD (Damaging)

As first contribution of this work, we point out that this claim is incorrect. In more detail, we present a nonce-misusing adversary that can forge a message for p-OMD in only 3

queries, including the forgery itself. At a high level, the attack relies on the observation that if an evaluation for p-OMD is made for a certain nonce, the adversary learns (most of) the corresponding state values. If the adversary is allowed to misuse the nonces, this means that it can effectively influence the state values, and henceforth generate a forgery. We also point out where the mistake occurs in the proof. We stress that this attack *does not* invalidate the security of p-OMD (nor OMD) in the nonce-respecting setting: that proof seems sound and the scheme achieves confidentiality and integrity.

1.2 Spoed (Simplifying)

One may argue that the flaw slipped into [19] in part due to the complex character of p-OMD. Indeed, the specification of p-OMD consists of 6 cases (or in fact 13, if you consider the scheme in full detail), depending on the number of associated data and message blocks. The forking of one scheme into a plurality of cases entails difficulties both on the theory side, leading to longer and more cumbersome proofs (which are incidentally harder to verify, as in the case above), and on the practical side, forcing less efficient and error-prone implementations. Additionally, it does not particularly facilitate an easy understanding and adoption of the scheme.

Driven by these conclusions and the potential that p-OMD offers for certain scenarios, we next explore the possibilities to generalize and simplify p-OMD. In more detail, we introduce Spoed,[1] a variant of p-OMD that aims to provide a higher level of simplicity at the same efficiency as p-OMD. In more detail, Spoed is an authenticated encryption mode that can use any keyed compression function $F_K : \{0,1\}^{2n} \to \{0,1\}^n$ for $n \geq 1$ as its underlying primitive. Spoed differs from p-OMD in the following aspects:

- Most importantly, Spoed uses a generalized padding scheme GPAD. It takes as input the associated data A and the message M, and injectively maps those to generalized message blocks of size $2n$ bits. As GPAD includes the length encodings of A and M as one of the generalized message blocks, it allows to give a unified description of the scheme: *one* scheme for all variants;
- p-OMD relies on Gray codes for case separation. Due to the usage of the generalized padding scheme, we can resort to the simpler-to-grasp powering-up approach [20] or word-based LFSR approach by Granger et al. [11]. Note that the usage of these approaches for state sizes larger than 128 bits has only been validated recently [11].

We prove that, assuming F_K is a sufficiently secure keyed compression function, Spoed redeems the security results of p-OMD in the nonce-respecting setting. To instantiate F_K, one can use the SHA256 or SHA512 compression functions. These functions have been the target of extensive cryptanalysis, and their security is well understood. They are also in wide use and efficient implementations of them can be found for practically any platform.

Spoed makes exactly the same number of compression function calls as p-OMD, except in the rare case where $a > m$ and $a + m$ is odd (in which case Spoed makes one extra compression function call due to the length encoding of A and M). We see this as a modest price to pay for achieving a scheme that (i) has a shorter and simpler description, making it easier to implement, (ii) requires less precomputational overhead, and (iii) has a proof of about 1/4th the size of the proof of p-OMD, making it easier to verify. The fact that Spoed has a slightly improved security bound can be seen as a bonus.

[1] The name is an acronym for "Simplified Pure OMD Encryption and Decryption."

1.3 Spoednic (Salvaging)

For the cases where nonce-misuse resistance is needed, we introduce Spoednic.[2] Spoednic is a variant of Spoed preserving integrity, up to the birthday bound, in the nonce-misuse scenario at the cost of one additional finite field multiplication per primitive call. Intuitively, the finite field multiplication is used to obfuscate the value XORed into the state, thus preventing an adversary from choosing a "convenient" value.

We prove that Spoednic inherits all security traits of Spoed, and has the added benefit of preserving the integrity against a nonce-reusing adversary. Surprisingly, the proof for this case leads to a better security bound than the flawed one claimed for p-OMD [19].

We stress that the reader should not take Spoednic as a recommendation for allowing the nonce to repeat. The question about who is responsible for dealing with the uniqueness of the nonce is debated in the cryptographic community. One side to this discussion believes that making sure the nonce is unique is an implementation matter while the other side believes that it should be dealt with by the algorithms designers. Both sides agree that a repeating nonce is an unwanted scenario, and the contribution of Spoednic is to allow for a graceful fail rather than a disastrous one in this unwanted event.

2 Security Model

Throughout, $n \geq 1$ denotes the state size. By $\oplus$ we denote the exclusive-or (XOR) operation, and by $\otimes$ or $\cdot$ finite field multiplication over 2^n. Concatenation is denoted using $\|$. Denote by $\{0,1\}^*$ the set of binary strings of arbitrary length and by $\{0,1\}^n$ the set of blocks of size n. Denote by $(\{0,1\}^n)^+$ the set of strings of length a *positive* multiple of n. For an arbitrary string X, $|X|$ denotes its length, and $\langle |X| \rangle_n$ denotes its encoding in $n \geq 1$ bits. By $\text{left}_n(X)$ (resp. $\text{right}_n(X)$) we denote its n leftmost (resp. rightmost) bits. We use little-endian notation, which means the three notations "bit position 0", "the rightmost bit", and "the least significant bit" all refer to the same bit.

In Sect. 2.1, we describe our model for the security of authenticated encryption. Then, we present some theoretical background on keyed compression functions in Sect. 2.2.

2.1 Authenticated Encryption

Let $\Pi = (\mathcal{E}, \mathcal{D})$ be an authenticated encryption scheme, where

$$\mathcal{E} : (K, N, A, M) \mapsto (C, T) \text{ and}$$
$$\mathcal{D} : (K, N, A, C, T) \mapsto M/\perp$$

are the encryption and decryption functions of Π. Let \$ be a random function that returns $(C,T) \xleftarrow{\$} \{0,1\}^{|M|} \times \{0,1\}^\tau$ on every new tuple (N, A, M). In other words, $\mathcal{E}$ and \$ have the same interface, but the latter outputs a uniformly randomly drawn ciphertext and tag for every new input.

An adversary $\mathcal{A}$ is a probabilistic algorithm that has access to one or more oracles $\mathcal{O}$, denoted $\mathcal{A}^{\mathcal{O}}$. By $\mathcal{A}^{\mathcal{O}} = 1$ we denote the event that $\mathcal{A}$, after interacting with $\mathcal{O}$, outputs 1. In below games, the adversaries have oracle access to $\mathcal{E}_K$ or its counterpart \$, and possibly $\mathcal{D}_K$. The key K is randomly drawn from $\{0,1\}^k$ at the beginning of the security experiment. We say that $\mathcal{A}$ is *nonce-respecting* (nr) if it never queries its

[2] The name is an acronym for "Simplified Pure OMD Encryption and Decryption with Nonce-misuse Integrity Conserved."

encryption oracle under the same nonce twice, and *nonce-misusing* (nm) if it is allowed to make multiple encryption queries with the same nonce. The security definitions below follow [1, 4, 10, 12, 13].

We define the advantage of $\mathcal{A}$ in breaking the confidentiality of Π as follows:

$$\mathbf{Adv}^{\mathrm{conf}}_{\Pi}(\mathcal{A}) = \left|\mathbf{Pr}\left(K \xleftarrow{\$} \{0,1\}^k, \mathcal{A}^{\mathcal{E}_K} = 1\right) - \mathbf{Pr}\left(\mathcal{A}^{\$} = 1\right)\right| .$$

For $\mathsf{n} \in \{\mathsf{nr}, \mathsf{nm}\}$, we denote by $\mathbf{Adv}^{\mathrm{conf}}_{\Pi}(\mathsf{n}, q, \ell, \sigma, t)$ the maximum advantage over all n-adversaries that make at most q queries, each of length at most ℓ generalized message blocks and together of length at most σ generalized message blocks, and that run in time t.

For integrity, we consider an adversary that tries to forge a ciphertext, which means that $\mathcal{D}_K$ ever returns a valid message (other than $\perp$) on input (N, A, C, T) and no previous encryption query $\mathcal{E}_K(N, A, M)$ returned (C, T) for any M. Formally:

$$\mathbf{Adv}^{\mathrm{int}}_{\Pi}(\mathcal{A}) = \mathbf{Pr}\left(K \xleftarrow{\$} \{0,1\}^k, \mathcal{A}^{\mathcal{E}_K, \mathcal{D}_K} \text{ forges}\right) .$$

For $\mathsf{n} \in \{\mathsf{nr}, \mathsf{nm}\}$, we denote by $\mathbf{Adv}^{\mathrm{int}}_{\Pi}(\mathsf{n}, q_{\mathcal{E}}, q_{\mathcal{D}}, \ell, \sigma, t)$ the maximum advantage over all n-adversaries that make at most $q_{\mathcal{E}}$ encryption and $q_{\mathcal{D}}$ decryption queries, each of length at most ℓ generalized message blocks and together of length at most σ generalized message blocks, and that run in time t. We remark that the nonce-respecting condition *only* applies to encryption queries: the adversary is always allowed to make decryption queries for "old" nonces, and to make an encryption query using a nonce which is already used in a decryption query before.

2.2 (Tweakable) Keyed Compression Function

Let $F : \{0,1\}^k \times \{0,1\}^{n+m} \to \{0,1\}^n$ be a keyed compression function. Denote by $\mathrm{Func}(\{0,1\}^{n+m}, \{0,1\}^n)$ the set of all compression functions from $n+m$ to n bits. We define the PRF security of F as

$$\mathbf{Adv}^{\mathrm{prf}}_{F}(\mathcal{A}) = \left|\begin{array}{l}\mathbf{Pr}\left(K \xleftarrow{\$} \{0,1\}^k, \mathcal{A}^{F_K} = 1\right) - \\ \qquad \mathbf{Pr}\left(R \xleftarrow{\$} \mathrm{Func}(\{0,1\}^{n+m}, \{0,1\}^n), \mathcal{A}^{R} = 1\right)\end{array}\right| .$$

We denote by $\mathbf{Adv}^{\mathrm{prf}}_{F}(q, t)$ the maximum advantage over all adversaries that make at most q queries and that run in time t.

A tweakable keyed compression function $\widetilde{F} : \{0,1\}^k \times \mathcal{T} \times \{0,1\}^{n+m} \to \{0,1\}^n$ takes as additional input a tweak $t \in \mathcal{T}$. Denote by $\widetilde{\mathrm{Func}}(\mathcal{T}, \{0,1\}^{n+m}, \{0,1\}^n)$ the set of all tweakable compression functions from $n+m$ to n bits, where the tweak inputs come from $\mathcal{T}$. Formally, a tweakable keyed compression function is equivalent to a keyed compression function with a larger input, but for our analysis it is more convenient to adopt dedicated notation. We define the tweakable PRF ($\widetilde{\mathrm{PRF}}$) security of $\widetilde{F}$ as

$$\mathbf{Adv}^{\widetilde{\mathrm{prf}}}_{\widetilde{F}}(\mathcal{A}) = \left|\begin{array}{l}\mathbf{Pr}\left(K \xleftarrow{\$} \{0,1\}^k, \mathcal{A}^{\widetilde{F}_K} = 1\right) - \\ \qquad \mathbf{Pr}\left(\widetilde{R} \xleftarrow{\$} \widetilde{\mathrm{Func}}(\mathcal{T}, \{0,1\}^{n+m}, \{0,1\}^n), \mathcal{A}^{\widetilde{R}} = 1\right)\end{array}\right| .$$

We denote by $\mathbf{Adv}^{\widetilde{\mathrm{prf}}}_{F}(q, t)$ the maximum advantage over all adversaries that make at most q queries and that run in time t.

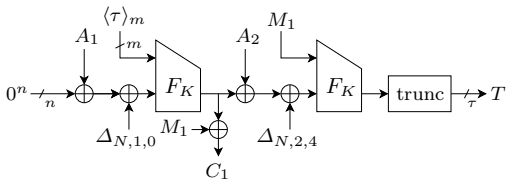

Fig. 1: p-OMD for the specific case of $|A| = 2n$ and $|M| = m$.

3 p-OMD

Let $k, m, n, \tau \in \mathbb{N}$ such that $m \leq n$. Let $F : \{0,1\}^k \times \{0,1\}^{n+m} \to \{0,1\}^n$ be a keyed compression function. p-OMD is a mapping that takes as input a key $K \in \{0,1\}^k$, a nonce $N \in \{0,1\}^{\leq n-1}$, an arbitrarily sized associated data $A \in \{0,1\}^*$, and an arbitrarily sized message $M \in \{0,1\}^*$, and it returns a ciphertext $C \in \{0,1\}^{|M|}$ and tag $T \in \{0,1\}^\tau$.

For our attack it suffices to describe p-OMD for the specific case where $|A| = 2n$ and $|M| = m$ (or in other words, the associated data consists of two integral blocks and the message of one integral block). It is depicted in Fig. 1 (and corresponds to Case A of [19]). Here,

$$\Delta_{N,1,0} = F_K(N\|10^{n-1-|N|}, 0^m) \oplus 16F_K(0^n, 0^m)\,,$$
$$\Delta_{N,2,4} = F_K(N\|10^{n-1-|N|}, 0^m) \oplus (32 \oplus 16 \oplus 4)F_K(0^n, 0^m)\,,$$

but our attack will not effectively use these masking values.

3.1 Preliminary Security Claims of p-OMD

In [19], Reyhanitabar et al. proved the following security levels for p-OMD:

Theorem 1. *We have*

$$\mathbf{Adv}^{\text{conf}}_{\text{p-OMD}}(\mathsf{nr}, q, \ell, \sigma, t) \leq \frac{3\sigma^2}{2^n} + \mathbf{Adv}^{\text{prf}}_F(2\sigma, t')\,,$$
$$\mathbf{Adv}^{\text{int}}_{\text{p-OMD}}(\mathsf{nr}, q_\mathcal{E}, q_\mathcal{D}, \ell, \sigma, t) \leq \frac{3\sigma^2}{2^n} + \frac{\ell q_\mathcal{D}}{2^n} + \frac{q_\mathcal{D}}{2^\tau} + \mathbf{Adv}^{\text{prf}}_F(2\sigma, t')\,,$$
$$\mathbf{Adv}^{\text{int}}_{\text{p-OMD}}(\mathsf{nm}, q_\mathcal{E}, q_\mathcal{D}, \ell, \sigma, t) \leq \frac{3\sigma^2}{2^n} + \frac{\ell(q_\mathcal{E}^2 + q_\mathcal{E})q_\mathcal{D}}{2^n} + \frac{q_\mathcal{D}}{2^\tau} + \mathbf{Adv}^{\text{prf}}_F(2\sigma, t')\,,$$

where $t' \approx t$.

In the updated version [18], the authors removed the last claim of the three as a result of this attack also presented in [2]. In the remainder of the section, we demonstrate why the bound does not hold.

3.2 Nonce-Misusing Attack on p-OMD

We consider a nonce-misusing adversary that operates as follows:

(i) Fix $N = \varepsilon$ and choose arbitrary $M \in \{0,1\}^m$ and $A_1, A_2, A_1' \in \{0,1\}^n$ such that $A_1 \neq A_1'$;
(ii) Query p-OMD$_K(N, A_1A_2, M) \to (C, T)$;
(iii) Query p-OMD$_K(N, A_1'A_2, M) \to (C', T')$;
(iv) Set $A_2' = C \oplus C' \oplus A_2$;

(v) Query forgery $\text{p-OMD}_K^{-1}(N, A_1'A_2', C', T)$.

For the first and second evaluation of p-OMD, it holds that the state difference right *before* the second F-evaluation equals $C \oplus C'$. The forgery is formed simply by adding this value to A_2. Consequently, it holds that the first call to p-OMD and the forgery attempt have the exact same input to the second F-evaluation, and thus the same tag. Therefore, the forgery attempt succeeds as

$$\text{p-OMD}_K^{-1}(N, A_1'A_2', C', T) = M$$

by construction. In other words, for some negligibly small t,

$$\mathbf{Adv}^{\text{int}}_{\text{p-OMD}}(\mathsf{nm}, 2, 1, 2, 6, t) = 1\,.$$

The issue appears in the proof of [19] in Lemma 4 case 4, and more specifically the analysis of probability $\mathbf{Pr}(\mathsf{intcol} \mid \mathsf{E}_4)$. The authors claim that an adversary can, indeed, find an internal collision, but that any such collision happens with a birthday bound only. This reasoning, however, assumes that the input to every F-call is random, which is not the case given that the adversary can re-use the nonce and thus observe and modify the state using encryption queries.

4 Spoed

We introduce the authenticated encryption scheme Spoed with the motivation of generalizing and simplifying p-OMD. As a bonus, the simplification allows for a better bound and a significantly shorter proof, making the scheme less susceptible to mistakes hiding in one of the lemmas.

4.1 Syntax

Let $k, n, \tau \in \mathbb{N}$ such that $\tau \leq n$. Here and throughout, we assume Spoed to process blocks of $m = n$ bits. However, the results easily generalize to arbitrary (but fixed) block sizes. Let $F : \{0,1\}^k \times \{0,1\}^{2n} \to \{0,1\}^n$ be a keyed compression function. Spoed consists of an encryption function $\mathcal{E}$ and a decryption function $\mathcal{D}$.

- The encryption function $\mathcal{E}$ takes as input a key $K \in \{0,1\}^k$, a nonce $N \in \{0,1\}^n$, an arbitrarily sized associated data $A \in \{0,1\}^*$, and an arbitrarily sized message $M \in \{0,1\}^*$. It returns a ciphertext $C \in \{0,1\}^{|M|}$ and a tag $T \in \{0,1\}^\tau$;
- The decryption function $\mathcal{D}$ takes as input a key $K \in \{0,1\}^k$, a nonce $N \in \{0,1\}^n$, an arbitrarily sized associated data $A \in \{0,1\}^*$, an arbitrarily sized ciphertext $C \in \{0,1\}^*$, and a tag $T \in \{0,1\}^\tau$. It returns either a message $M \in \{0,1\}^{|C|}$ such that M satisfies $\mathcal{E}(K, N, A, M) = (C, T)$ or a dedicated failure sign $\perp$.

The encryption and decryption function are required to satisfy

$$\mathcal{D}(K, N, A, \mathcal{E}(K, N, A, M)) = M$$

for any K, N, A, M.

4.2 Generalized Padding

Spoed uses a generalized padding function

$$\text{GPAD}_{n,\tau} : \{0,1\}^* \times \{0,1\}^* \to \left(\{0,1\}^{2n}\right)^+ .$$

It is indexed by state sizes n, τ, and it maps the associated data and message to generalized message blocks. Formally, it is defined as follows: First, A (associated data) and X (message or ciphertext) are padded into n-bit message blocks $A_1 \| \cdots \| A_a = A \| 0^{n-|A| \bmod n}$ and $X_1 \| \cdots \| X_m = X \| 0^{n-|X| \bmod n}$, respectively. Denote $\ell = \max\left\{m, \lceil \frac{a+m}{2} \rceil\right\} + 1$, and define $\mathrm{len}(A, X) = \langle |A| \rangle_{n/2} \| \langle |X| \rangle_{n/2}$.[3] The function $\mathrm{GPAD}_{n,\tau}(A, X)$ outputs $Z_1, \ldots, Z_\ell$ as follows:

if $a \leq m$:	if $a > m$, $a+m$ even:	if $a > m$, $a+m$ odd:
$Z_1 = \langle \tau \rangle_n \parallel A_1$	$Z_1 = \langle \tau \rangle_n \parallel A_1$	$Z_1 = \langle \tau \rangle_n \parallel A_1$
$Z_2 = X_1 \parallel A_2$	$Z_2 = X_1 \parallel A_2$	$Z_2 = X_1 \parallel A_2$
$\ldots$	$\ldots$	$\ldots$
$Z_a = X_{a-1} \parallel A_a$	$Z_{m+1} = X_m \parallel A_{m+1}$	$Z_{m+1} = X_m \parallel A_{m+1}$
$Z_{a+1} = X_a \parallel 0^n$	$Z_{m+2} = A_{m+2} \parallel A_{m+3}$	$Z_{m+2} = A_{m+2} \parallel A_{m+3}$
$\ldots$	$\ldots$	$\ldots$
$Z_{\ell-1} = X_{m-1} \parallel 0^n$	$Z_{\ell-1} = A_{a-2} \parallel A_{a-1}$	$Z_{\ell-1} = A_{a-1} \parallel A_a$
$Z_\ell = X_m \parallel \mathrm{len}(A, X)$	$Z_\ell = A_a \parallel \mathrm{len}(A, X)$	$Z_\ell = 0^n \parallel \mathrm{len}(A, X)$

The encoding of the message length is included in order to circumvent the need for a case distinction in the description of Spoed. Note that, in fact, almost any injective padding rule would do the job; however, for our purposes the described $\mathrm{GPAD}_{n,\tau}$ is the most suitable. We generically write $Z_i = Z_i^0 \parallel Z_i^1$, and denote $Z^\beta = Z_1^\beta \| \cdots \| Z_\ell^\beta$ for $\beta \in \{0, 1\}$.

4.3 Data Processing

Spoed is designed with the SHA256 and SHA512 compression functions in mind as its underlying primitive. SHA256 is a compression function

$$\mathrm{SHA256} : \{0,1\}^{256} \times \{0,1\}^{512} \to \{0,1\}^{256} .$$

Similarly, SHA512 is a compression function $\mathrm{SHA512} : \{0,1\}^{512} \times \{0,1\}^{1024} \to \{0,1\}^{512}$. In the sequel, we will define Spoed using SHA256, or in other words used keyed compression function

$$F_K(Z) = \mathrm{SHA256}(K, Z) ,$$

where K is injected through the chaining value interface, and the block is injected through the message interface. Note that this implicitly means that we take $k = n = 256$. We nevertheless continue to use k and n for clarity. Note that Spoed can be equivalently designed using the SHA512 compression function, but a proper change in the sizes of Z_i^0 and Z_i^1 should be introduced.

We now informally describe how to use Spoed, and refer the reader to Algorithms 1 and 2 for a formal specification. Define $L = F_K(N \| 0)$. First, the associated data and message are padded into

$$(Z_1, \ldots, Z_\ell) = \mathrm{GPAD}_{n,\tau}(A, M) ,$$

where each Z_i is a $(2n = 512)$-bit block consisting of two blocks $Z_i = Z_i^0 \parallel Z_i^1$ of size 256 bits each. Spoed reads all blocks but the last one sequentially and processes them by

$$t_i = F_K(t_{i-1} \oplus 2^i L \oplus Z_i^0 \parallel Z_i^1) ,$$

[3] As we show in Sect. 6, Spoed achieves birthday bound security, and the limitation of the length of X and A to $2^{n/2} - 1$ does not pose any issues.

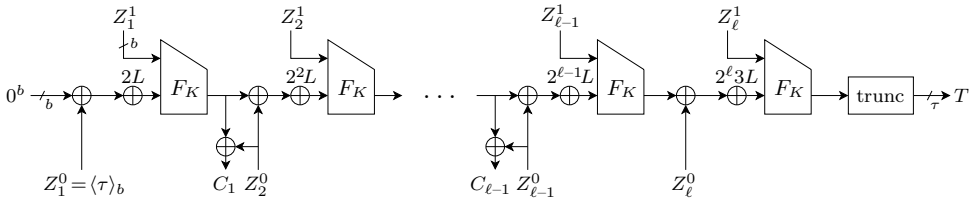

Fig. 2: Spoed encryption, which outputs $C = \text{left}_{|M|}(C_1 \| \cdots \| C_{\ell-1})$ and T. Here, $L = F_K(N\|0)$

where $t_0 = 0^n$. The ciphertext block C_i is generated as $C_i = t_i \oplus M_i$, chopped to the appropriate length if M_i does not contain a full amount of n message bits. The last block Z_ℓ contains the lengths of the message and the associated data and is processed through

$$t_\ell = F_K(t_{\ell-1} \oplus 2^\ell 3L \oplus Z_\ell^0 \parallel Z_\ell^1)\,.$$

The tag T is generated by removing the leftmost 256-τ bits of t_ℓ. Spoed is depicted in Fig. 2.

Decryption goes fairly the same way: a t_i and a C_i value are used to recover M_i, and the state is set to C_i. This eventually leads to a verification tag T', and the message M is released if $T = T'$.

Algorithm 1 Spoed encryption $\mathcal{E}$

Input: (K, N, A, M)
Output: (C, T)
$(Z_1, \ldots, Z_\ell) = \text{GPAD}_{n,\tau}(A, M)$
$m = \lceil |M|/n \rceil$
$L = F_K(N\|0)$
$t_0 = 0^n$
for $i = 1, \ldots, \ell - 1$ **do**
  $t_i = F_K(t_{i-1} \oplus 2^i L \oplus Z_i^0 \parallel Z_i^1)$
  $C_i = t_i \oplus Z_{i+1}^0$
$t_\ell = F_K(t_{\ell-1} \oplus 2^\ell 3L \oplus Z_\ell^0 \parallel Z_\ell^1)$
$C = \text{left}_{|M|}(C_1 \| \cdots \| C_{\ell-1})$
$T = \text{left}_\tau(t_\ell)$
return (C, T)

Algorithm 2 Spoed decryption $\mathcal{D}$

Input: (K, N, A, C, T)
Output: M or $\perp$
$(Z_1, \ldots, Z_\ell) = \text{GPAD}_{n,\tau}(A, C)$
$m = \lceil |C|/n \rceil$, $\rho = |C| \bmod n$
$L = F_K(N\|0)$
$t_0 = 0^n$, $M_0 = Z_1^0$
for $i = 1, \ldots, \ell - 1$ **do**
  $t_i = F_K(t_{i-1} \oplus 2^i L \oplus M_{i-1} \parallel Z_i^1)$
  if $i < m$ **then** $M_i = t_i \oplus Z_{i+1}^0$
  if $i = m$ **then** $M_i = \text{left}_\rho(t_i) \oplus Z_{i+1}^0$
  if $i > m$ **then** $M_i = Z_{i+1}^0$
$t_\ell = F_K(t_{\ell-1} \oplus 2^\ell 3L \oplus Z_\ell^0 \parallel Z_\ell^1)$
$M = \text{left}_{|C|}(M_1 \| \cdots \| M_{\ell-1})$
$T' = \text{left}_\tau(t_\ell)$
return $T = T'$? $M : \perp$

4.4 Security of Spoed

Spoed achieves confidentiality and integrity against nonce-respecting adversaries. Note that we do not claim security against nonce-misusing adversaries.

Theorem 2. *We have*

$$\mathbf{Adv}^{\text{conf}}_{\text{Spoed}}(\mathsf{nr}, q, \ell, \sigma, t) \le \frac{1.5\sigma^2}{2^n} + \mathbf{Adv}^{\text{prf}}_F(2\sigma, t')\,,$$

$$\mathbf{Adv}^{\text{int}}_{\text{Spoed}}(\mathsf{nr}, q_\mathcal{E}, q_\mathcal{D}, \ell, \sigma, t) \le \frac{1.5\sigma^2}{2^n} + \frac{\ell q_\mathcal{D}}{2^n} + \frac{q_\mathcal{D}}{2^\tau} + \mathbf{Adv}^{\text{prf}}_F(2\sigma, t')\,,$$

where $t' \approx t$.

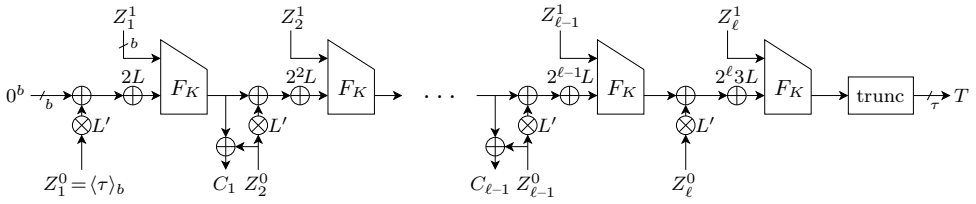

Fig. 3: Spoednic encryption, which outputs $C = \text{left}_{|M|}(C_1 \| \cdots \| C_{\ell-1})$ and T. Here, $L = F_K(N\|0)$ and $L' = F_K(N\|1)$

These bounds, surprisingly, improve over the ones of p-OMD (see Thm. 1), but with a much shorter proof. The proof is given in Sect. 6.

5 Spoednic

Spoed is simpler and more efficient than p-OMD, but it also falls victim to nonce-misuse attacks. In this section, we introduce Spoednic, a strengthened version of Spoed that retains some level of security if the nonce gets reused. As a matter of fact, Spoednic differs from Spoed in (and only in) the fact that it uses an additional subkey $L' = F_K(N\|1)$ and that the input values Z_i^0 are blinded by L'. More formally, Spoednic inherits the syntax and generalized padding from Spoed (see Sect. 4.1 and Sect. 4.2). The data processing is fairly similar to that of Spoed (Sect. 4.3); we only present the depiction in Fig. 3 and the formal description in Algorithms 3 and 4. Both algorithms differ from Algorithms 1 and 2 *only in* lines 3 and 6. Spoednic boils down to Spoed (Fig. 2) if one would use $L' = 1$ instead of $L' = F_K(N\|1)$.

Algorithm 3 Spoednic encryption $\mathcal{E}$

Input: (K, N, A, M)
Output: (C, T)
1: $(Z_1, \ldots, Z_\ell) = \text{GPAD}_{n,\tau}(A, M)$
2: $m = \lceil |M|/n \rceil$
3: $L = F_K(N\|0)$, $L' = F_K(N\|1)$
4: $t_0 = 0^n$
5: **for** $i = 1, \ldots, \ell - 1$ **do**
6: $\quad t_i = F_K(t_{i-1} \oplus 2^i L \oplus (Z_i^0 \cdot L') \parallel Z_i^1)$
7: $\quad C_i = t_i \oplus Z_{i+1}^0$
8: $t_\ell = F_K(t_{\ell-1} \oplus 2^\ell 3L \oplus Z_\ell^0 \parallel Z_\ell^1)$
9: $C = \text{left}_{|M|}(C_1 \| \cdots \| C_{\ell-1})$
10: $T = \text{left}_\tau(t_\ell)$
11: **return** (C, T)

Algorithm 4 Spoednic decryption $\mathcal{D}$

Input: (K, N, A, C, T)
Output: M or $\perp$
1: $(Z_1, \ldots, Z_\ell) = \text{GPAD}_{n,\tau}(A, C)$
2: $m = \lceil |C|/n \rceil$, $\rho = |C| \bmod n$
3: $L = F_K(N\|0)$, $L' = F_K(N\|1)$
4: $t_0 = 0^n$, $M_0 = Z_1^0$
5: **for** $i = 1, \ldots, \ell - 1$ **do**
6: $\quad t_i = F_K(t_{i-1} \oplus 2^i L \oplus (M_{i-1} \cdot L') \parallel Z_i^1)$
7: $\quad$ **if** $i < m$ **then** $M_i = t_i \oplus Z_{i+1}^0$
8: $\quad$ **if** $i = m$ **then** $M_i = \text{left}_\rho(t_i) \oplus Z_{i+1}^0$
9: $\quad$ **if** $i > m$ **then** $M_i = Z_{i+1}^0$
10: $t_\ell = F_K(t_{\ell-1} \oplus 2^\ell 3L \oplus Z_\ell^0 \parallel Z_\ell^1)$
11: $M = \text{left}_{|C|}(M_1 \| \cdots \| M_{\ell-1})$
12: $T' = \text{left}_\tau(t_\ell)$
13: **return** $T = T'$? $M : \perp$

5.1 Security of Spoednic

We prove that Spoednic achieves confidentiality against nonce-respecting adversaries and integrity against both nonce-respecting and nonce-misusing adversaries. Note that we do not claim confidentiality against nonce-misusing adversaries.

Theorem 3. *We have*

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{conf}}_{\mathrm{Spoednic}}(\mathsf{nr}, q, \ell, \sigma, t) &\leq \frac{1.5\sigma^2}{2^n} + \mathbf{Adv}^{\mathrm{prf}}_{F}(3\sigma, t')\,, \\
\mathbf{Adv}^{\mathrm{int}}_{\mathrm{Spoednic}}(\mathsf{nr}, q_{\mathcal{E}}, q_{\mathcal{D}}, \ell, \sigma, t) &\leq \frac{1.5\sigma^2}{2^n} + \frac{\ell q_{\mathcal{D}}}{2^n} + \frac{q_{\mathcal{D}}}{2^\tau} + \mathbf{Adv}^{\mathrm{prf}}_{F}(3\sigma, t')\,, \\
\mathbf{Adv}^{\mathrm{int}}_{\mathrm{Spoednic}}(\mathsf{nm}, q_{\mathcal{E}}, q_{\mathcal{D}}, \ell, \sigma, t) &\leq \frac{1.5\sigma^2}{2^n} + \frac{\ell q_{\mathcal{E}}^2/2}{2^n} + \frac{\ell q_{\mathcal{E}} q_{\mathcal{D}}}{2^n} + \frac{q_{\mathcal{D}}}{2^\tau} + \mathbf{Adv}^{\mathrm{prf}}_{F}(3\sigma, t')\,,
\end{aligned}$$

where $t' \approx t$.

The proof is given in Sect. 7.

6 Security of Spoed (Theorem 2)

The proof of Thm. 2 is given in Sect. 6.2. It relies on a preliminary result on a tweakable keyed compression function, which will be given in Sect. 6.1.

6.1 Security of Tweakable Keyed Compression Function

In the proof of Spoed we will use the following result. It is in fact an abstraction of the XE tweakable blockcipher [20] to compression functions, and it has also been used for OMD [9] and p-OMD [18], albeit with a worse bound.

Lemma 1. *Let* $F : \{0,1\}^k \times \{0,1\}^{2n} \to \{0,1\}^n$ *be a keyed compression function. Let* $\mathcal{T} = [1, 2^{n/2}] \times [0,1] \times \{0,1\}^n$*, and define* $\widetilde{F} : \{0,1\}^k \times \mathcal{T} \times \{0,1\}^{2n} \to \{0,1\}^n$ *as*

$$\widetilde{F}(K, (\alpha, \beta, N), S) = F(K, (2^\alpha 3^\beta \cdot F_K(N\|0) \parallel 0^n) \oplus S)\,. \tag{1}$$

Then, we have

$$\mathbf{Adv}^{\widetilde{\mathrm{prf}}}_{\widetilde{F}}(q, t) \leq \frac{1.5q^2}{2^n} + \mathbf{Adv}^{\mathrm{prf}}_{F}(2q, t')\,,$$

where $t' \approx t$.

Proof. The proof is performed using the H-coefficient technique [7,16]. It closely follows the proof of [11, Thm. 2]; the only significant differences appear in the fact that the underlying primitive is a one-way function instead of a permutation, and hence various bad events have become redundant. To wit, in the terminology of [11, Thm. 2], the events $\mathrm{bad}_{1,2}$ and $\mathrm{bad}_{2,K}$ are inapplicable (as the adversary has no access to the underlying primitive), and for the events $\mathrm{bad}_{1,1}$, $\mathrm{bad}_{1,K}$, and $\mathrm{bad}_{K,K}$, we only have to consider input collisions to the primitives. Checking the corresponding bounds reveals a term $1.5q^2/2^n$.

As a first step, we replace the evaluations of F_K for $K \xleftarrow{\$} \{0,1\}^k$ by a random function $R : \{0,1\}^{2n} \to \{0,1\}^n$. As every evaluation of $\widetilde{F}$ renders at most 2 evaluations of F_K, this step costs us $\mathbf{Adv}^{\mathrm{prf}}_{F}(2q, t')$, where $t' \approx t$, and allows us to consider

$$\widetilde{F} : ((\alpha, \beta, N), S) \mapsto R((2^\alpha 3^\beta \cdot R(N\|0) \parallel 0^n) \oplus S)\,, \tag{2}$$

based on $R \xleftarrow{\$} \mathrm{Func}(\{0,1\}^{2n}, \{0,1\}^n)$. As we have replaced the underlying function F by a secret random primitive, we can focus on adversaries with unbounded computational power, and consider them to be information theoretic. Without loss of generality, any such adversary is deterministic. For the remainder of the analysis, consider any fixed

deterministic adversary $\mathcal{A}$. Without loss of generality, we assume that $\mathcal{A}$ does not repeat any queries.

Let $R \xleftarrow{\$} \mathrm{Func}(\{0,1\}^{2n}, \{0,1\}^n)$ and $\widetilde{R} \xleftarrow{\$} \widetilde{\mathrm{Func}}(\mathcal{T}, \{0,1\}^{2n}, \{0,1\}^n)$. Consider any fixed deterministic adversary $\mathcal{A}$. In the real world, it has access to $\widetilde{F}$ of (2), while in the ideal world it has access to $\widetilde{R}$, and its goal is to distinguish both worlds. It makes q queries to the oracle, which are summarized in a view

$$\nu_F = \{(\alpha_1, \beta_1, N_1, S_1, T_1), \ldots, (\alpha_q, \beta_q, N_q, S_q, T_q)\}\,.$$

Note that, as $\mathcal{A}$ is deterministic, this view ν_F properly summarizes the interaction with the oracle. To suit the analysis, we will provide $\mathcal{A}$ with additional information *after* its interaction with its oracle. In more detail, it is given a *subkey transcript* ν_L that includes the computations of $R(N\|0)$ for all $N \in \{N_1, \ldots, N_q\}$. As the latter set may include duplicates, i.e., it may be that $N_i = N_j$, the formalism of ν_L requires some notation. Let $\{M_1, \ldots, M_r\}$ be a minimal set that includes $N_1, \ldots, N_q$. Then, after the interaction of $\mathcal{A}$ with its oracle, we reveal

$$\nu_L = \{(M_1, L_1), \ldots, (M_r, L_r)\}\,,$$

In the real world the values $L_1, \ldots, L_r$ are defined as $L_i = R(M_i\|0)$, while in the ideal world, these values are randomly generated dummy subkeys $L_i \xleftarrow{\$} \{0,1\}^n$. Clearly, the disclosure of ν_L is without loss of generality as it only increases the adversary's chances. The complete view is defined as $\nu = (\nu_F, \nu_L)$. It is important to note that, as $\mathcal{A}$ never repeats queries, ν_F does not contain any duplicate elements. Neither does ν_L, by minimality of the set $\{M_1, \ldots, M_r\}$.

H-Coefficient Technique. For brevity, denote $\mathcal{A}$'s distinguishing advantage by $\Delta_{\mathcal{A}}(\widetilde{F}; \widetilde{R})$. Denote by $X_{\widetilde{F}}$ the probability distribution of views when $\mathcal{A}$ is interacting with $\widetilde{F}$ and by $X_{\widetilde{R}}$ the probability distribution of views when $\mathcal{A}$ is interacting with $\widetilde{R}$. Let $\mathcal{V}$ be the set of all attainable views, being the views that can be generated from $\widetilde{R}$ with non-zero probability. Let $\mathcal{V} = \mathcal{V}_{\mathrm{good}} \cup \mathcal{V}_{\mathrm{bad}}$ be a partition of the set of attainable views. The H-coefficient technique states the following. Let $0 \le \varepsilon \le 1$ be such that for all $\nu \in \mathcal{V}_{\mathrm{good}}$ we have

$$\frac{\mathbf{Pr}\left(X_{\widetilde{F}} = \nu\right)}{\mathbf{Pr}\left(X_{\widetilde{R}} = \nu\right)} \ge 1 - \varepsilon\,.$$

Then, the distinguishing advantage of $\mathcal{A}$ satisfies

$$\Delta_{\mathcal{A}}(\widetilde{F}; \widetilde{R}) \le \varepsilon + \mathbf{Pr}\left(X_{\widetilde{R}} \in \mathcal{V}_{\mathrm{bad}}\right)\,. \tag{3}$$

We refer to [6] for a proof.

Bad Transcripts. Note that every tuple in ν_F uniquely fixes a subkey in ν_L and therewith uniquely fixes one evaluation $R(s) = t$. On the other hand, the evaluations in ν_L represent evaluations of R themselves. Informally, we will consider a transcript as *bad* if there exist two different tuples that have the same input to R. Formally, we say that a view ν is *bad* if it satisfies one of the following conditions:

Bad1. There exist $(\alpha, \beta, N, S, T) \in \nu_F$ and $(N, L), (M^*, L^*) \in \nu_L$ such that:

$$(2^\alpha 3^\beta \cdot L \parallel 0^n) \oplus S = M^* \parallel 0^n\,;$$

Bad2. There exist distinct $(\alpha, \beta, N, S, T), (\alpha^*, \beta^*, N^*, S^*, T^*) \in \nu_F$ and (not necessarily distinct) $(N, L), (N^*, L^*) \in \nu_L$ such that:

$$(2^\alpha 3^\beta \cdot L \parallel 0^n) \oplus S = (2^{\alpha^*} 3^{\beta^*} \cdot L^* \parallel 0^n) \oplus S^* \,.$$

Probability of Bad Transcripts. Consider a view ν in the ideal world $\widetilde{R}$. We will consider both bad events separately.

Bad1. Consider any query $(\alpha, \beta, N, S, T) \in \nu_F$ with corresponding subkey $(N, L) \in \nu_L$, and let $(M^*, L^*) \in \nu_L$ (q^2 choices in total). The queries render a bad view if

$$2^\alpha 3^\beta \cdot L = S^0 \oplus M^* \,.$$

As in the ideal world $L \xleftarrow{\$} \{0,1\}^n$, this equation is satisfied with probability $1/2^n$. Summing over all possible choices of queries, Bad1 is satisfied with probability at most $q^2/2^n$;

Bad2. Consider any distinct $(\alpha, \beta, N, S, T), (\alpha^*, \beta^*, N^*, S^*, T^*) \in \nu_F$ with corresponding $(N, L), (N^*, L^*) \in \nu_L$ ($\binom{q}{2}$ choices in total). The queries render a bad view if

$$2^\alpha 3^\beta \cdot L \oplus S^0 = 2^{\alpha^*} 3^{\beta^*} \cdot L^* \oplus S^{*0} \wedge S^1 = S^{*1} \,.$$

Clearly, if $N \neq N^*$, then $L \xleftarrow{\$} \{0,1\}^n$ is generated independently of the remaining values, and the first part of the condition holds with probability $1/2^n$. Similar for the case where $N = N^*$ but $2^\alpha 3^\beta \neq 2^{\alpha^*} 3^{\beta^*}$. On the other hand, if $N = N^*$ and $2^\alpha 3^\beta =^{\alpha^*} 3^{\beta^*}$, we necessarily have $(N, \alpha, \beta) = (N^*, \alpha^*, \beta^*)$ (due to the non-colliding property of $2^\alpha 3^\beta$). As the two queries in ν_F are distinct, we have $S \neq S^*$, making above condition false. Concluding, Bad2 is satisfied with probability at most $\binom{q}{2}/2^n$.

We thus obtained that $\mathbf{Pr}\left(X_{\widetilde{R}} \in \mathcal{V}_{\text{bad}}\right) \leq 1.5q^2/2^n$.

Good Transcripts. Consider a good view ν. Denote by $\Omega_{\widetilde{F}}$ the set of all possible oracles in the real world and by $\text{comp}_{\widetilde{F}}(\nu) \subseteq \Omega_{\widetilde{F}}$ the set of oracles compatible with view ν. Define $\Omega_{\widetilde{R}}$ and $\text{comp}_{\widetilde{R}}(\nu)$ similarly. The probabilities $\mathbf{Pr}\left(X_{\widetilde{F}} = \nu\right)$ and $\mathbf{Pr}\left(X_{\widetilde{R}} = \nu\right)$ can be computed as follows:

$$\mathbf{Pr}\left(X_{\widetilde{F}} = \nu\right) = \frac{|\text{comp}_{\widetilde{F}}(\nu)|}{|\Omega_{\widetilde{F}}|} \text{ and } \mathbf{Pr}\left(X_{\widetilde{R}} = \nu\right) = \frac{|\text{comp}_{\widetilde{R}}(\nu)|}{|\Omega_{\widetilde{R}}|} \,.$$

Note that $|\Omega_{\widetilde{F}}| = (2^n)^{2^{2n}}$ and $|\Omega_{\widetilde{R}}| = (2^n)^{|\mathcal{T}|+2^{2n}} \cdot (2^n)^r$ (taking into account that in the ideal world ν contains r dummy subkeys). The computation of the number of compatible oracles is a bit more technical. Starting with $\text{comp}_{\widetilde{F}}(\nu)$, as ν is a good view, every tuple in ν represents *exactly one* evaluation of R, $q + r$ in total, and hence the number of functions R compatible with ν is $|\text{comp}_{\widetilde{F}}(\nu)| = (2^n)^{2^{2n}-(q+r)}$. Next, for $\text{comp}_{\widetilde{R}}(\nu)$, the tuples in ν_F all define *exactly one* evaluation of $\widetilde{R}$, q in total, and ν_L fixes all dummy keys. Therefore, the number of compatible oracles in the ideal world is $|\text{comp}_{\widetilde{R}}(\nu)| = (2^n)^{|\mathcal{T}|+2^{2n}-q}$. We consequently obtain

$$\frac{\mathbf{Pr}\left(X_{\widetilde{F}} = \nu\right)}{\mathbf{Pr}\left(X_{\widetilde{R}} = \nu\right)} = \frac{|\text{comp}_{\widetilde{F}}(\nu)| \cdot |\Omega_{\widetilde{R}}|}{|\Omega_{\widetilde{F}}| \cdot |\text{comp}_{\widetilde{R}}(\nu)|} = \frac{(2^n)^{2^{2n}-(q+r)} \cdot (2^n)^{|\mathcal{T}|+2^{2n}} \cdot (2^n)^r}{(2^n)^{2^{2n}} \cdot (2^n)^{|\mathcal{T}|+2^{2n}-q}} = 1 \,,$$

putting $\varepsilon = 0$.

Conclusion. The proof is concluded via (3) and above computations. □

Note that p-OMD uses tweaks of the form 2^α, while we use $2^\alpha 3^\beta$. This is not a problem as long as the offsets are unique [20] (i.e., there is no $(\alpha,\beta) \neq (\alpha',\beta')$ such that $2^\alpha 3^\beta = 2^{\alpha'} 3^{\beta'}$). For the case of $n = 128$, Rogaway [20] proved—via the computation of discrete logarithms—that the tweak domain $[1, 2^{n/2}] \times [0, 1]$ works properly, but this result is inadequate for our purposes as we use a compression function with $n \in \{256, 512\}$. Granger et al. [11] recently computed discrete logarithms for $n \leq 1024$, therewith confirming properness of the tweak set domain. Note that the tweak sets computed in [11,20] commonly exclude the all-zero tweak $(\alpha,\beta) = (0,0)$ because it is a representative of 1 and hence problematic for XEX: see also [20, Sect. 6] and [15, Sect. 4]. Because F is a one-way function, its security analysis follows the one of XE, and this issue does not apply.

Also from an efficiency point of view, there is a difference between the masking of $\widetilde{F}$ in p-OMD and in Spoed. In more detail, p-OMD uses the Gray code masking (also used in OCB1 and OCB3) while for Spoed we have opted to describe it with powering-up (used in OCB2 and in various CAESAR candidates). Krovetz and Rogaway demonstrated that Gray codes are more efficient than powering-up [14], but on the downside they require more precomputation. Granger et al. [11] revisited the principle of masking of tweakable blockciphers, and presented a masking technique based on word-based linear feedback shift registers that improves over both Gray codes and powering-up in terms of efficiency and simplicity. The new masking technique can be implemented with Spoed with no sacrifice in security (and the result of Lem. 1 still applies).

6.2 Proof of Theorem 2

Let $K \in \{0,1\}^k$. Note that all evaluations of F_K are done in a tweakable manner, namely via (1). We replace these tweakable evaluations of F_K by a random tweakable compression function $\widetilde{R} \xleftarrow{\$} \widetilde{\mathrm{Func}}([1, 2^{n/2}] \times [0,1] \times \{0,1\}^n, \{0,1\}^{2n}, \{0,1\}^n)$. Note that for both confidentiality and integrity, the underlying $\widetilde{F}_K$ is invoked at most σ times. In other words, this step costs (cf. Lem. 1)

$$\mathbf{Adv}^{\widetilde{\mathrm{prf}}}_{\widetilde{F}}(\sigma, t) \leq \frac{1.5\sigma^2}{2^n} + \mathbf{Adv}^{\mathrm{prf}}_{F}(2\sigma, t')\,,$$

where $t' \approx t$. This step has led us to an idealized version of Spoed, called IdSpoed. IdSpoed is depicted in Fig. 4. Concretely, we have obtained that

$$\mathbf{Adv}^{\mathrm{conf}}_{\mathrm{Spoed}}(\mathsf{nr}, q, \ell, \sigma, t) \leq \mathbf{Adv}^{\mathrm{conf}}_{\mathrm{IdSpoed}}(\mathsf{nr}, q, \ell, \sigma) + \frac{1.5\sigma^2}{2^n} + \mathbf{Adv}^{\mathrm{prf}}_{F}(2\sigma, t')\,,$$

$$\mathbf{Adv}^{\mathrm{int}}_{\mathrm{Spoed}}(\mathsf{nr}, q_{\mathcal{E}}, q_{\mathcal{D}}, \ell, \sigma, t) \leq \mathbf{Adv}^{\mathrm{int}}_{\mathrm{IdSpoed}}(\mathsf{nr}, q_{\mathcal{E}}, q_{\mathcal{D}}, \ell, \sigma) + \frac{1.5\sigma^2}{2^n} + \mathbf{Adv}^{\mathrm{prf}}_{F}(2\sigma, t')\,,$$

where t dropped out of the advantage function for IdSpoed because it has become irrelevant (formally, we proceed by considering an adversary that is unbounded in time). We prove in Lem. 2 that its confidentiality security satisfies $\mathbf{Adv}^{\mathrm{conf}}_{\mathrm{IdSpoed}}(\mathsf{nr}, q, \ell, \sigma) = 0$, and in Lem. 3 that it provides integrity up to bound $\mathbf{Adv}^{\mathrm{int}}_{\mathrm{IdSpoed}}(\mathsf{nr}, q_{\mathcal{E}}, q_{\mathcal{D}}, \ell, \sigma) \leq \frac{\ell q_{\mathcal{D}}}{2^n} + \frac{q_{\mathcal{D}}}{2^\tau}$.

Lemma 2. *The advantage of any nonce-respecting adversary trying to break the confidentiality of* IdSpoed *is bounded as:*

$$\mathbf{Adv}^{\mathrm{conf}}_{\mathrm{IdSpoed}}(\mathsf{nr}, q, \ell, \sigma) = 0\,.$$

Proof. The functions $\widetilde{R}^N_{i,j}$ for $i = 1, \ldots, \ell-1$, $j = 0, 1$, and $N \in \{0,1\}^n$ are independently and randomly distributed compression functions. As the adversary is assumed to be

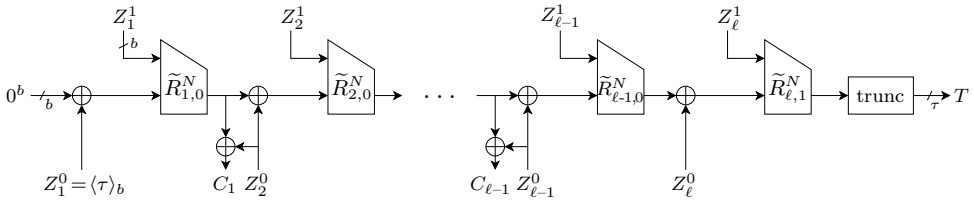

Fig. 4: IdSpoed encryption, which outputs $C = \mathrm{left}_{|M|}(C_1 \| \cdots \| C_{\ell-1})$ and T

nonce-respecting, every nonce is used at most once. Every nonce is used in at most ℓ calls to $\widetilde{R}$, but these calls are by design all for different tweaks $(i, j) \in [1, 2^{n/2}] \times [0, 1]$. Therefore, all responses are randomly generated from $\{0,1\}^n$, and all ciphertext blocks and tag values are perfectly random. □

Lemma 3. *The advantage of any nonce-respecting adversary trying to break the integrity of* IdSpoed *is bounded as:*

$$\mathbf{Adv}^{\mathrm{int}}_{\mathrm{IdSpoed}}(\mathsf{nr}, q_{\mathcal{E}}, q_{\mathcal{D}}, \ell, \sigma) \leq \frac{\ell q_{\mathcal{D}}}{2^n} + \frac{q_{\mathcal{D}}}{2^\tau}\,.$$

Proof. Assume that $\mathcal{A}$ has made encryption queries (N^j, A^j, M^j) for $j = 1, \ldots, q_{\mathcal{E}}$, and denote the ciphertexts and tags by (C^j, T^j). Write $(Z_1^j, \ldots, Z_{\ell^j}^j) = \mathrm{GPAD}_{n,\tau}(A^j, M^j)$ and denote the in- and outputs of the random functions by (s_i^j, t_i^j) for $i = 1, \ldots, \ell^j$.

Consider any forgery attempt (N, A, C, T), and denote its length by ℓ. Denote the message computed upon decryption by M. Refer to the state values as (s_i, t_i) for $i = 1, \ldots, \ell$, and write $(Z_1, \ldots, Z_\ell) = \mathrm{GPAD}_{n,\tau}(A, M)$. The forgery is successful if $T = \mathrm{left}_\tau(t_\ell)$.

Denote by col the event that there exists an encryption query j with $N^j = N$, $\ell^j = \ell$, and an index $i \in \{1, \ldots, \ell\}$, such that

$$t_{i-1}^j \oplus Z_i^{0j} \parallel Z_i^{1j} \neq t_{i-1} \oplus Z_i^0 \parallel Z_i^1 \ \wedge\ t_i^j = t_i\,.$$

Note that, as the adversary is nonce-respecting, there is at most one query j with $N^j = N$. We have, using shorthand notation $[i = \ell]$ for 0 if $i \neq \ell$ and 1 if $i = \ell$,

$$\mathbf{Pr}\,(\mathsf{col}) \leq \sum_{i=1}^{\ell} \mathbf{Pr}\left(s_i^j \neq s_i \wedge \widetilde{R}^N_{i,[i=\ell]}(s_i^j) = \widetilde{R}^N_{i,[i=\ell]}(s_i)\right) \leq \frac{\ell}{2^n}\,. \tag{4}$$

We make the following, fairly simple, case distinction:

(i) $N \notin \{N^1, \ldots, N^{q_{\mathcal{E}}}\}$. This particularly means that $\widetilde{R}$ has never been queried for tweak $(\ell, 1, N)$, and thus that $\widetilde{R}^N_{\ell,1}$ responds with $t_\ell \xleftarrow{\$} \{0,1\}^n$. The forgery is successful with probability $1/2^\tau$;

(ii) $N = N^j$ for some (unique) j. As the different evaluations of IdSpoed for different tweaks are independent, it suffices to focus on these two construction queries (the jth encryption query and the forgery). We proceed with a further case distinction:
   - $\ell \neq \ell^j$. This, again, means that $\widetilde{R}$ has never been queried for tweak $(\ell, 1, N)$. The forgery is successful with probability $1/2^\tau$;
   - $\ell = \ell^j$. We proceed with a further case distinction:
     - $s_\ell \neq s_{\ell^j}^j$. In this case, $\widetilde{R}$ has been queried before for tweak $(\ell, 1, N)$, but only once (as the adversary must be nonce-respecting) and never on input s_ℓ. Consequently, the response t_ℓ is uniformly randomly drawn from $\{0,1\}^n$ and the forgery is successful with probability $1/2^\tau$;

- $s_\ell = s^j_{\ell j}$. As the forgery must be different from the encryption queries, and as $\mathrm{GPAD}_{n,\tau}$ is an injective mapping, this case implies the existence of a non-trivial state collision. Hence, the forgery is successful with probability at most $\mathbf{Pr}\left(\mathsf{col}\right)$.

Concluding, the forgery is successful with probability at most $\mathbf{Pr}\left(\mathsf{col}\right) + 1/2^\tau$, where $\mathbf{Pr}\left(\mathsf{col}\right)$ is bounded in (4). A summation over all $q_{\mathcal{D}}$ forgery attempts (cf. [3]) gives our final bound. □

7 Security of Spoednic (Theorem 3)

The proof of Thm. 3 is given in Sect. 7.2. It relies on a preliminary result on a tweakable keyed compression function, which will be given in Sect. 7.1.

7.1 Security of Tweakable Keyed Compression Function

We will use a slightly more complex version of the tweakable keyed compression function of Sect. 6.1, where the masking using $Z^0_i \cdot L'$ is included within the function. The proof is a fairly straightforward extension of the one of Lem. 1.

Lemma 4. *Let $F : \{0,1\}^k \times \{0,1\}^{2n} \to \{0,1\}^n$ be a keyed compression function. Let $\mathcal{T} = [1, 2^{n/2}] \times [0,1] \times \{0,1\}^n \times \{0,1\}^n$, and define $\widetilde{F} : \{0,1\}^k \times \mathcal{T} \times \{0,1\}^{2n} \to \{0,1\}^n$ as*

$$\widetilde{F}(K, (\alpha, \beta, A, N), S) = F(K, (2^\alpha 3^\beta \cdot F_K(N\|0) \oplus A \cdot F_K(N\|1) \parallel 0^n) \oplus S)\,. \tag{5}$$

Then, we have

$$\mathbf{Adv}^{\widetilde{\mathrm{prf}}}_{\widetilde{F}}(q,t) \leq \frac{1.5q^2}{2^n} + \mathbf{Adv}^{\mathrm{prf}}_F(3q, t')\,,$$

where $t' \approx t$.

Proof. The proof is a slight extension of the one of Lem. 1, where now we have twice as many subkeys. We only sketch the major differences.

Again, the first step is the replacement of F_K for $K \xleftarrow{\$} \{0,1\}^k$ by a random function $R : \{0,1\}^{2n} \to \{0,1\}^n$. As every query to $\widetilde{F}$ renders at most 3 evaluations of F, this step costs us $\mathbf{Adv}^{\mathrm{prf}}_F(3q,t')$, where $t' \approx t$, and allows us to consider

$$\widetilde{F} : ((\alpha, \beta, A, N), S) \mapsto R((2^\alpha 3^\beta \cdot R(N\|0) \oplus A \cdot R(N\|1) \parallel 0^n) \oplus S)\,, \tag{6}$$

based on $R \xleftarrow{\$} \mathrm{Func}(\{0,1\}^{2n}, \{0,1\}^n)$. The movement to information theoretic, deterministic, adversary $\mathcal{A}$ goes as before.

Let $R \xleftarrow{\$} \mathrm{Func}(\{0,1\}^{2n}, \{0,1\}^n)$ and $\widetilde{R} \xleftarrow{\$} \widetilde{\mathrm{Func}}(\mathcal{T}, \{0,1\}^{2n}, \{0,1\}^n)$. Consider any fixed deterministic adversary $\mathcal{A}$. In the real world, it has access to $\widetilde{F}$ of (6), while in the ideal world it has access to $\widetilde{R}$, and its goal is to distinguish both worlds. It makes q queries to the oracle, which are summarized in a view

$$\nu_F = \{(\alpha_1, \beta_1, A_1, N_1, S_1, T_1), \ldots, (\alpha_q, \beta_q, A_q, N_q, S_q, T_q)\}\,.$$

As an extension to the proof of Lem. 1, we now reveal to the adversary two subkey transcripts ν_L and $\nu_{L'}$, the former captures the evaluations $R(N\|0)$ and the latter the evaluations $R(N\|1)$ for all $N \in \{N_1, \ldots, N_q\}$. More formally, let $\{M_1, \ldots, M_r\}$ be a

minimal set that includes $N_1, \ldots, N_q$. Then, after the interaction of $\mathcal{A}$ with its oracle, we reveal

$$\nu_L = \{(M_1, L_1), \ldots, (M_r, L_r)\}\,,$$
$$\nu_{L'} = \{(M_1, L'_1), \ldots, (M_r, L'_r)\}\,.$$

In the real world, the subkeys are generated as $L_i = R(M_i\|0)$ and $L'_i = R(M_i\|1)$, while in the ideal world they are randomly generated dummy subkeys $L_i, L'_i \xleftarrow{\$} \{0,1\}^n$. The complete view is defined as $\nu = (\nu_F, \nu_L, \nu_{L'})$.

Bad Transcripts. Formally, we say that a view ν is *bad* if it satisfies one of the following conditions:

Bad1. There exist $(\alpha, \beta, A, N, S, T) \in \nu_F$, $(N, L) \in \nu_L$, $(N, L') \in \nu_{L'}$, and $(M^*, L^*) \in \nu_L \cup \nu_{L'}$ such that:

$$(2^\alpha 3^\beta \cdot L \oplus A \cdot L' \parallel 0^n) \oplus S = M^* \parallel 0^n \vee M^* \parallel 1^n\,;$$

Bad2. There exist distinct $(\alpha, \beta, A, N, S, T), (\alpha^*, \beta^*, A^*, N^*, S^*, T^*) \in \nu_F$, $(N, L), (N^*, L^*) \in \nu_L$, and $(N, L'), (N^*, L'^*) \in \nu_{L'}$ such that:

$$(2^\alpha 3^\beta \cdot L \oplus A \cdot L' \parallel 0^n) \oplus S = (2^{\alpha^*} 3^{\beta^*} \cdot L^* \oplus A^* \cdot L'^* \parallel 0^n) \oplus S^*\,.$$

Probability of Bad Transcripts. Consider a view ν in the ideal world $\widetilde{R}$. We will consider both bad events separately.

Bad1. Consider any query $(\alpha, \beta, A, N, S, T) \in \nu_F$ with corresponding subkeys $(N, L) \in \nu_L$ and $(N, L') \in \nu_{L'}$, and let $(M^*, L^*) \in \nu_L \cup \nu_{L'}$. Note that we have q choices for the query from ν_F, and q possible values M^* (even though $\nu_L \cup \nu_{L'}$ may contain up to $2q$ tuples). The queries render a bad view if

$$2^\alpha 3^\beta \cdot L \oplus A \cdot L' = S^0 \oplus M^*\,.$$

As in the ideal world $L \xleftarrow{\$} \{0,1\}^n$, this equation is satisfied with probability $1/2^n$. Summing over all possible choices of queries, Bad1 is satisfied with probability at most $q^2/2^n$;

Bad2. Consider any distinct $(\alpha, \beta, A, N, S, T), (\alpha^*, \beta^*, A^*, N^*, S^*, T^*) \in \nu_F$, $(N, L), (N^*, L^*) \in \nu_L$, and $(N, L'), (N^*, L'^*) \in \nu_{L'}$ ($\binom{q}{2}$ choices in total). The queries render a bad view if

$$2^\alpha 3^\beta \cdot L \oplus A \cdot L' \oplus S^0 = 2^{\alpha^*} 3^{\beta^*} \cdot L^* \oplus A \cdot L'^* \oplus S^{*0} \wedge S^1 = S^{*1}\,.$$

The case $N \neq N^*$ and the case $N = N^*$ but $2^\alpha 3^\beta \neq 2^{\alpha^*} 3^{\beta^*}$ are as in Lem. 1. Similarly, if $N = N^*$ but $A \neq A^*$, we can rely on the randomness of L' to argue that the condition is satisfied with probability $1/2^n$. On the other hand, if $(\alpha, \beta, A, N) = (\alpha^*, \beta^*, A^*, N^*)$, this necessarily implies that $S \neq S^*$, making above condition false. Concluding, Bad2 is satisfied with probability at most $\binom{q}{2}/2^n$.

We thus obtained that $\mathbf{Pr}\left(X_{\widetilde{R}} \in \mathcal{V}_{\text{bad}}\right) \leq 1.5q^2/2^n$.

Good Transcripts. The analysis is fairly identical to the one of Lem. 1, and henceforth omitted.

Conclusion. The proof is concluded via (3) and above computations. □

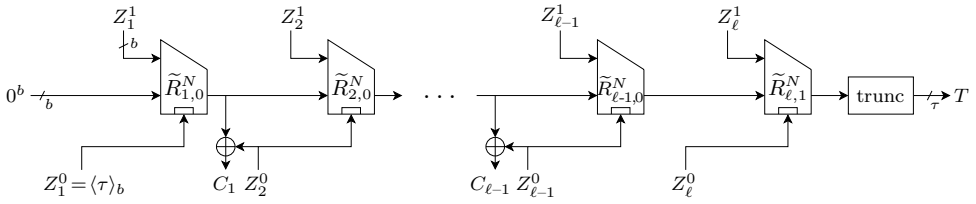

Fig. 5: IdSpoednic encryption, which outputs $C = \mathrm{left}_{|M|}(C_1 \| \cdots \| C_{\ell-1})$ and T. The boxes in $\widetilde{R}$ indicate that Z_i^0 also functions as a tweak

7.2 Proof of Theorem 3

Let $K \in \{0,1\}^k$. Note that all evaluations of F_K are done in a tweakable manner, namely via (5). We replace these tweakable evaluations of F_K by a random tweakable compression function $\widetilde{R} \xleftarrow{\$} \widetilde{\mathrm{Func}}([1, 2^{n/2}] \times [0,1] \times \{0,1\}^n \times \{0,1\}^n, \{0,1\}^{2n}, \{0,1\}^n)$. Note that for both confidentiality and integrity, the underlying $\widetilde{F}_K$ is invoked at most σ times. In other words, this step costs (cf. Lem. 4)

$$\mathbf{Adv}^{\widetilde{\mathrm{prf}}}_{\widetilde{F}}(\sigma, t) \leq \frac{1.5\sigma^2}{2^n} + \mathbf{Adv}^{\mathrm{prf}}_{F}(3\sigma, t')\,,$$

where $t' \approx t$. This step has led us to an idealized version of Spoednic, called IdSpoednic. IdSpoednic is depicted in Fig. 5. Concretely, we have obtained that

$$\mathbf{Adv}^{\mathrm{conf}}_{\mathrm{Spoednic}}(\mathsf{nr}, q, \ell, \sigma, t) \leq \mathbf{Adv}^{\mathrm{conf}}_{\mathrm{IdSpoednic}}(\mathsf{nr}, q, \ell, \sigma) + \frac{1.5\sigma^2}{2^n} + \mathbf{Adv}^{\mathrm{prf}}_{F}(3\sigma, t')\,,$$

$$\mathbf{Adv}^{\mathrm{int}}_{\mathrm{Spoednic}}(\mathsf{n}, q_{\mathcal{E}}, q_{\mathcal{D}}, \ell, \sigma, t) \leq \mathbf{Adv}^{\mathrm{int}}_{\mathrm{IdSpoednic}}(\mathsf{n}, q_{\mathcal{E}}, q_{\mathcal{D}}, \ell, \sigma) + \frac{1.5\sigma^2}{2^n} + \mathbf{Adv}^{\mathrm{prf}}_{F}(3\sigma, t')\,,$$

where $\mathsf{n} \in \{\mathsf{nr}, \mathsf{nm}\}$, and where t dropped out of the advantage function for IdSpoednic because it has become irrelevant. The remainder of the proof centers around this scheme. For the nonce-respecting setting, the bounds of Lem. 2 and Lem. 3 carry over almost verbatim, with the same security bound. We consider integrity in the nonce-misuse setting in Lem. 5 and prove that $\mathbf{Adv}^{\mathrm{int}}_{\mathrm{IdSpoednic}}(\mathsf{nm}, q_{\mathcal{E}}, q_{\mathcal{D}}, \ell, \sigma) \leq \frac{\ell q_{\mathcal{E}}^2/2}{2^n} + \frac{\ell q_{\mathcal{E}} q_{\mathcal{D}}}{2^n} + \frac{q_{\mathcal{D}}}{2^\tau}$.

Lemma 5. *The advantage of any nonce-misusing adversary trying to break the integrity of* IdSpoednic *is bounded as:*

$$\mathbf{Adv}^{\mathrm{int}}_{\mathrm{IdSpoed}}(\mathsf{nm}, q_{\mathcal{E}}, q_{\mathcal{D}}, \ell, \sigma) \leq \frac{\ell q_{\mathcal{E}}^2/2}{2^n} + \frac{\ell q_{\mathcal{E}} q_{\mathcal{D}}}{2^n} + \frac{q_{\mathcal{D}}}{2^\tau}\,.$$

Proof. At a high level, the proof follows the one of Lem. 3, with the difference that now, potentially, nonces may be the same. This leads to a slightly worse pre-forgery-attempt bounding of the collision probability from $\frac{\ell q_{\mathcal{D}}}{2^n}$ to $\frac{\ell q_{\mathcal{E}} q_{\mathcal{D}}}{2^n}$, as well as to an addition of the term $\frac{\ell q_{\mathcal{E}}^2/2}{2^n}$ to the final bound, which accounts for the number of inner collision in different encryption queries under the same nonce.

Assume that $\mathcal{A}$ has made encryption queries (N^j, A^j, M^j) for $j = 1, \ldots, q_{\mathcal{E}}$, and denote the ciphertexts and tags by (C^j, T^j). Write $(Z_1^j, \ldots, Z_{\ell^j}^j) = \mathrm{GPAD}_{n,\tau}(A^j, M^j)$ and denote the in- and outputs of the random functions by (s_i^j, t_i^j) for $i = 1, \ldots, \ell^j$.

Denote by $\mathsf{col}\mathcal{E}$ the event that there exist two distinct encryption queries j, j' with $N^j = N^{j'}$, and an index $i \in \{1, \ldots, \ell^j\}$, such that

$$t_{i-1}^j \,\|\, Z_i^{1j} \neq t_{i-1}^{j'} \,\|\, Z_i^{1j'} \;\wedge\; t_i^j = t_i^{j'}\,.$$

We have, using shorthand notation $[i = \ell]$ for 0 if $i \neq \ell$ and 1 if $i = \ell$,

$$\mathbf{Pr}\left(\mathsf{col}\mathcal{E}\right) \leq \sum_{\substack{j,j'=1 \\ j \neq j'}}^{q_\mathcal{E}} \sum_{i=1}^{\min\{\ell^j,\ell^{j'}\}} \mathbf{Pr}\left(s_i^j \neq s_i^{j'} \wedge \widetilde{R}^N_{i,[i=\ell^j]}(Z_i^{0j}, s_i^j) = \widetilde{R}^N_{i,[i=\ell^{j'}]}(Z_i^{0j'}, s_i^{j'})\right) \leq \frac{\ell\binom{q_\mathcal{E}}{2}}{2^n}\,. \tag{7}$$

The remainder of the analysis is under the assumption that $\neg\mathsf{col}\mathcal{E}$ applies, and we add the term of (7) at the end.

Consider any forgery attempt (N, A, C, T), and denote its length by ℓ. Denote the message computed upon decryption by M. Refer to the state values as (s_i, t_i) for $i = 1, \ldots, \ell$, and write $(Z_1, \ldots, Z_\ell) = \mathrm{GPAD}_{n,\tau}(A, M)$. The forgery is successful if $T = \mathrm{left}_\tau(t_\ell)$.

Denote by $\mathsf{col}\mathcal{D}$ the event that there exists an encryption query j with $N^j = N$, $\ell^j = \ell$, and an index $i \in \{1, \ldots, \ell\}$, such that

$$t_{i-1}^j \parallel Z_i^{1j} \neq t_{i-1} \parallel Z_i^1 \ \wedge\ t_i^j = t_i\,.$$

We have

$$\mathbf{Pr}\left(\mathsf{col}\mathcal{D} \mid \neg\mathsf{col}\mathcal{E}\right) \leq \sum_{j=1}^{q_\mathcal{E}} \sum_{i=1}^{\ell} \mathbf{Pr}\left(s_i^j \neq s_i \wedge \widetilde{R}^N_{i,[i=\ell]}(s_i^j) = \widetilde{R}^N_{i,[i=\ell]}(s_i)\right) \leq \frac{\ell q_\mathcal{E}}{2^n}\,. \tag{8}$$

We make the following, fairly simple, case distinction:

(i) $N \notin \{N^1, \ldots, N^{q_\mathcal{E}}\}$. This particularly means that $\widetilde{R}$ has never been queried for tweak $(\ell, 1, Z_\ell^0, N)$, and thus that $\widetilde{R}^N_{\ell,1}(Z_\ell^0, \cdot)$ responds with $t_\ell \xleftarrow{\$} \{0,1\}^n$. The forgery is successful with probability $1/2^\tau$;

(ii) $N = N^j$ for $j \in \{1, \ldots, q'_\mathcal{E}\}$ for some $1 \leq q'_\mathcal{E} \leq q_\mathcal{E}$. Note that we have implicitly reordered the encryption queries such that the ones for nonce N are the first $q'_\mathcal{E}$. This is without loss of generality, as the different evaluations of IdSpoednic for different tweaks are independent. We proceed with a further case distinction:
   - $\ell \notin \{\ell^1, \ldots, \ell^{q'_\mathcal{E}}\}$. This, again, means that $\widetilde{R}$ has never been queried for tweak $(\ell, 1, Z_\ell^0, N)$. The forgery is successful with probability $1/2^\tau$;
   - $\ell = \ell^j$ for $j \in \{1, \ldots, q''_\mathcal{E}\}$ for some $1 \leq q''_\mathcal{E} \leq q'_\mathcal{E}$. Note that we have implicitly reordered the encryption queries such that the ones for nonce N and length ℓ are the first $q''_\mathcal{E}$. This is, again, without loss of generality. We proceed with a further case distinction:
     - $s_\ell \notin \{s^1_{\ell^1}, \ldots, s^{q_\mathcal{E}}_{\ell^{q''_\mathcal{E}}}\}$. In this case, $\widetilde{R}$ has been queried before for tweak $(\ell, 1, *, N)$, where $*$ denotes any tweak input Z_ℓ^{0j} which is left irrelevant, but never on input s_ℓ. Consequently, the response t_ℓ is uniformly randomly drawn from $\{0,1\}^n$ and the forgery is successful with probability $1/2^\tau$;
     - $s_\ell = s^j_{\ell^j}$ for some $j \in \{1, \ldots, q''_\mathcal{E}\}$. As the forgery must be different from the encryption queries, as $\mathrm{GPAD}_{n,\tau}$ is an injective mapping, and moreover as $\neg\mathsf{col}\mathcal{E}$, this case implies the existence of a non-trivial state collision. Hence, the forgery is successful with probability at most $\mathbf{Pr}\left(\mathsf{col}\mathcal{D} \mid \neg\mathsf{col}\mathcal{E}\right)$.

Concluding, the forgery is successful with probability at most $\mathbf{Pr}\left(\mathsf{col} \mid \neg\mathsf{col}\mathcal{E}\right) + 1/2^\tau$, where $\mathbf{Pr}\left(\mathsf{col} \mid \neg\mathsf{col}\mathcal{E}\right)$ is bounded in (8). A summation over all $q_\mathcal{D}$ forgery attempts (cf. [3]) gives our final bound. □

Acknowledgments. This work was supported in part by the Research Council KU Leuven: GOA TENSE (GOA/11/007). In addition, this work was partially supported by

the Research Fund KU Leuven, OT/13/071, and by European Unions Horizon 2020 research and innovation programme under No H2020-MSCA-ITN-2014-643161 ECRYPT-NET. Bart Mennink is a Postdoctoral Fellow of the Research Foundation – Flanders (FWO).

References

1. 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, Part I. Lecture Notes in Computer Science, vol. 8269, pp. 424–443. Springer (2013)
2. Ashur, T., Mennink, B.: Trivial nonce-misusing attack on pure OMD. Cryptology ePrint Archive, Report 2015/175 (2015)
3. Bellare, M., Goldreich, O., Mityagin, A.: The power of verification queries in message authentication and authenticated encryption. Cryptology ePrint Archive, Report 2004/309 (2004)
4. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. J. Cryptology 21(4), 469–491 (2008)
5. CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness (May 2014), `http://competitions.cr.yp.to/caesar.html`
6. Chen, S., Lampe, R., Lee, J., Seurin, Y., Steinberger, J.P.: Minimizing the two-round even-mansour cipher. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. Lecture Notes in Computer Science, vol. 8616, pp. 39–56. Springer (2014)
7. Chen, S., Steinberger, J.P.: Tight security bounds for key-alternating ciphers. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. Lecture Notes in Computer Science, vol. 8441, pp. 327–350. Springer (2014)
8. Cogliani, S., Maimut, D., Naccache, D., do Canto, R.P., Reyhanitabar, R., Vaudenay, S., Vizár, D.: Offset Merkle-Damgrd (OMD) version 1.0 (2014), submission to CAESAR competition
9. Cogliani, S., Maimut, D., Naccache, D., do Canto, R.P., Reyhanitabar, R., Vaudenay, S., Vizár, D.: OMD: A Compression Function Mode of Operation for Authenticated Encryption. In: Joux, A., Youssef, A.M. (eds.) SAC 2014. Lecture Notes in Computer Science, vol. 8781, pp. 112–128. Springer (2014)
10. Fleischmann, E., Forler, C., Lucks, S.: McOE: A family of almost foolproof on-line authenticated encryption schemes. In: Canteaut, A. (ed.) FSE 2012. Lecture Notes in Computer Science, vol. 7549, pp. 196–215. Springer (2012)
11. Granger, R., Jovanovic, P., Mennink, B., Neves, S.: Improved masking for tweakable blockciphers with applications to authenticated encryption. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part I. Lecture Notes in Computer Science, vol. 9665, pp. 263–293. Springer (2016)
12. Iwata, T., Ohashi, K., Minematsu, K.: Breaking and repairing GCM security proofs. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. Lecture Notes in Computer Science, vol. 7417, pp. 31–49. Springer (2012)
13. Jovanovic, P., Luykx, A., Mennink, B.: Beyond $2^{c/2}$ security in sponge-based authenticated encryption modes. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part I. Lecture Notes in Computer Science, vol. 8873, pp. 85–104. Springer (2014)
14. Krovetz, T., Rogaway, P.: The software performance of authenticated-encryption modes. In: Joux, A. (ed.) FSE 2011. Lecture Notes in Computer Science, vol. 6733, pp. 306–327. Springer (2011)
15. Minematsu, K.: Improved security analysis of XEX and LRW modes. In: Biham, E., Youssef, A.M. (eds.) SAC 2006. Lecture Notes in Computer Science, vol. 4356, pp. 96–113. Springer (2006)
16. Patarin, J.: The "coefficients H" technique. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. Lecture Notes in Computer Science, vol. 5381, pp. 328–345. Springer (2008)
17. Reyhanitabar, R., Vaudenay, S., Vizár, D.: Misuse-Resistant Variants of the OMD Authenticated Encryption Mode. In: Chow, S.S.M., Liu, J.K., Hui, L.C.K., Yiu, S. (eds.) Provable Security 2014. Lecture Notes in Computer Science, vol. 8782, pp. 55–70. Springer (2014)

18. Reyhanitabar, R., Vaudenay, S., Vizár, D.: Boosting OMD for almost free authentication of associated data. In: Leander, G. (ed.) FSE 2015. Lecture Notes in Computer Science, vol. 9054, pp. 411–427. Springer (2015)
19. Reyhanitabar, R., Vaudenay, S., Vizár, D.: Boosting OMD for almost free authentication of associated data. FSE 2015 preprint version (2015)
20. Rogaway, P.: Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In: Lee, P.J. (ed.) ASIACRYPT 2004. Lecture Notes in Computer Science, vol. 3329, pp. 16–31. Springer (2004)