Generic Security of the Ascon Mode: On the Power of Key Blinding

Charlotte Lefevre and Bart Mennink

Digital Security Group, Radboud University, Nijmegen, The Netherlands
charlotte.lefevre@ru.nl, b.mennink@cs.ru.nl

Abstract. The Ascon authenticated encryption scheme has recently been selected as winner of the NIST Lightweight Cryptography competition. Despite its fame, a comprehensive and self-contained generic security treatment of its mode remains absent. In this work we present a thorough security analysis of the Ascon mode: we consider multi-user and possibly nonce-misuse security by default, but more importantly, we particularly investigate the role of the key blinding. More technically, our analysis includes an authenticity study in various attack settings. This analysis includes a description of a security model of authenticity under state recovery, that captures the idea that the mode aims to still guarantee authenticity and security against key recovery even if an inner state is revealed to the adversary in some way, for instance through leakage. We prove that Ascon satisfies this security property, thanks to its unique key blinding technique.

Keywords: Ascon · NIST LWC · authenticated encryption · key blinding · security under state recovery.

1 Introduction

The area of lightweight cryptography has been a focal point of research in the last decades. The initial research focused merely on the design of lightweight block ciphers, such as NOEKEON [19], PRESENT [13], and GIFT [3]. The introduction of sponge functions [8] has given rise to a large amount of hash functions that aimed to be lightweight, such as QUARK [2], PHOTON [28], and SPONGENT [12]. In more recent years, the quest mostly centered around authenticated encryption. This research was mainly driven by the CAESAR competition for authenticated encryption design [14] and thereafter by the lightweight cryptography competition organized by the US National Institute of Standards and Technology (NIST) [36]. In the CAESAR competition, the Ascon authenticated encryption scheme [20,22] was selected as winner in the category lightweight. In the NIST lightweight cryptography competition, Ascon [21,22] was selected as overall winner, this time consisting of an authenticated encryption scheme and a hash function, out of 57 submissions.

Ascon is a permutation-based authenticated encryption scheme that is inspired by the duplex construction [10,18,34]. It is based on a 320-bit permutation

p. It initializes its 320-bit state as the concatenation of an initialization vector, a 128-bit key, and a 128-bit nonce. It permutes the state using p and it compresses the key again into the state. Then, Ascon proceeds with the absorption of associated data in a keyed-sponge-style fashion [1,8,11,35] and the encryption of plaintext in a duplex-style fashion [10,18], both with a round-reduced version of the permutation p (the details are irrelevant for current discussion). Finally, the state is blinded using the key, permuted using p, and 128-bits of its output are once more blinded using the key and output as tag. A detailed description of the mode can be found in Section 3.

The mode of Ascon resembles ideas of the authenticated encryption modes SpongeWrap [10] or MonkeySpongeWrap [33], but not quite. The most crucial difference is the key blinding at the beginning and at the end to achieve extra robustness against state recovery. These key blindings are absent in existing sponge-/duplex-based authenticated encryption security proofs. (Additionally, a difference between Ascon and SpongeWrap/MonkeySpongeWrap is the slightly different domain separators, but this difference is minor.) This difference makes it impossible to argue security of the Ascon mode based on the multi-user security bounding of the duplex of Daemen et al. [18] and Dobraunig and Mennink [24]. There exists a dedicated security proof of a sponge-/duplex-based mode that seems to get close to the security of the Ascon mode, which is the security proof of Jovanovic et al. [30, 31] for NORX. In detail, the authors claim that their security proof carries over if there is key blinding at the end. However, *no argument* is given to support that claim. At the same time, Jovanovic et al. adopted a by now outdated multicollision bounding technique (refer to [16, Section 4.3] and [33, Section 4.2] for a discussion about different techniques).

This leaves us with a quite unsatisfiable state of affairs: the winner of both the CAESAR competition (in the category lightweight) and the NIST lightweight cryptography competition *does not have a thorough self-contained generic security proof for its mode*. In an independent work, Chakraborty et al. [15] performed a first analysis of the Ascon mode, but only considered plain nonce-based security in the single-user setting.

On top of that, and more importantly, the role of additional key blindings as done in Ascon is not well-understood from a theoretical perspective. To be precise, the designers of Ascon have chosen to include the key blindings in their scheme for security reasons: they claim that these extra key blindings allow the scheme to achieve authenticity even under state recovery. This attack setting, for example, applies to the case of leveled implementations [37], where the outer permutations get stronger protection than the (round-reduced) inner permutation. In this case, nonce-misusing attackers may get a higher chance at recovering the state of an inner permutation call. This security property *has never been formally explored nor analyzed*.

1.1 Our Contribution

We present a rigorous analysis of the Ascon authenticated encryption mode, with particular attention to the additional key blinding. In a nutshell, we demonstrate

that the mode achieves (i) multi-user nonce-based confidentiality and authenticity, (ii) multi-user authenticity even under nonce misuse (though with a slightly worse bound), and (iii) multi-user authenticity even under state recovery.

The first result (i) is in the conventional model for multi-user authenticated encryption, and demonstrate that the Ascon mode achieves nonce-based confidentiality up to time complexity around $\min\{2^k/\mu, 2^{n/2}, 2^c\}$, where k is the key size, μ the number of users, n the permutation size, and c the sponge capacity, and nonce-based authenticity up to time complexity around $\min\{2^k/\mu, 2^n/M_{\mathcal{E}}, 2^c/M_{\mathcal{D}}\}$, where $M_{\mathcal{E}}$ and $M_{\mathcal{D}}$ are the encryption and decryption complexity bounds. In comparison, Chakraborty et al. [15] independently proved security up to a time complexity of around $\min\{2^k, 2^c, 2^b/M\}$, where $M = M_{\mathcal{E}} + M_{\mathcal{D}}$: this is a tighter bound, but only in the single-user setting, and our findings in result (i) complement well. The difference in tightness comes from the use of a different proof technique: in our paper, (i) merely serves as starter for the more important results (ii) and (iii), for which a different proof technique appeared to be more favorable.

If the adversary reuses nonces, result (ii), confidentiality is not guaranteed anymore, but we demonstrate that the scheme still achieves authenticity up to time complexity around $\min\{2^k/\mu, 2^c/(M_{\mathcal{E}} + M_{\mathcal{D}})\}$. The proofs are inspired by the proof of Jovanovic et al. [30] for NORX, but significantly differ in the fact that the key blinding is taken into account, and that a more modern approach to bound multicollisions is adopted, namely the technique of Choi et al. [17], but for which we found a slight improvement (see Lemma 6 in Appendix A).

The third result (iii) is in a setting that was never formally explored in the first place. The only earlier work in this direction is by Guo et al. in Theorem 4 of [26], the full version of [27]. However, this result is informal (as explicitly mentioned by the authors), and a closer look at the reasoning reveals that it contains several incomplete and incorrect steps.[1] Therefore, we first formalize the right model that captures the idea that the scheme still achieves authenticity under state recovery. This model is rather subtle as it gives the adversary some kind of power to choose when and which secret state to obtain. However, we observe that a model very similar to muCIML2 of [26, Definition 2], but adapted to our formalism and notation, suffices. Then, we demonstrate that Ascon *without key blinding* (i.e., with only using the key to initialize the state) does *not* achieve authenticity under state recovery, but that the actual Ascon *with key blinding* achieves this unique property. The security bound is, logically, worse than that of plain authenticity, but is still reasonably high: authenticity under state recovery is achieved up to time complexity around $\min\{2^k/\mu, 2^{c/2}\}$. We also argue tightness of the security result.

[1] In fact, we derived our result independently, and even discarded the idea behind the reasoning of [26] as it would not allow us to obtain a tight security bound. We elaborate on this in Section 5.4, after the proof of Theorem 4.

1.2 Outline

We start with preliminaries in Section 2. We describe the Ascon mode in Section 3, and the authenticated encryption security model in Section 4. In particular, in Section 4.3 we define and argue our security model of authenticity under state recovery. Security of the Ascon mode is derived in Section 5, with confidentiality in Section 5.1, authenticity in nonce-misuse and nonce-respecting respectively in Section 5.2 and Section 5.3, and authenticity under state recovery in Section 5.4. We conclude in Section 6.

2 Preliminaries

Let $m, n \in \mathbb{N}$ with $m \leq n$. We denote $[\![m, n]\!] = \{m, \ldots, n\}$. We denote by $\{0,1\}^n$ the set of n-bit strings and $\{0,1\}^*$ the set of arbitrarily long strings. The empty string is denoted by ϵ. For $X \in \{0,1\}^*$ we denote by $\mathrm{pad}_n(X)$ the function that splits X into n-bit blocks where the last block is of size between 0 and $n-1$ bits. We denote by $\mathrm{pad}_n^{10^*}(X)$ the function that pads X with a 1 and a sufficient number of 0s so that the length of X becomes a multiple of n bits, and that subsequently splits it into blocks of n bits.

For $a \leq |X|$, we denote by $\lceil x \rceil_a$ to denote the leftmost a bits and $\lfloor x \rfloor_a$ as the rightmost a bits of x. For $X, Y \in \{0,1\}^*$, we denote their concatenation by $X \parallel Y$ and if $|X| = |Y|$ their bitwise exclusive or by $X \oplus Y$. In addition, if $c \leq \min\{|X|, |Y|\}$, $X \overset{c}{=} Y$ means that $\lfloor X \rfloor_c = \lfloor Y \rfloor_c$. For a set $\mathcal{S}$ of elements, we write $X \overset{c}{\in} \mathcal{S}$ to mean that $X \overset{c}{=} Y$ for some $Y \in \mathcal{S}$. We stretch the notation even further, and state that for two sets $\mathcal{S}$ and $\mathcal{T}$ we have $\mathcal{S} \overset{c}{\cap} \mathcal{T} \neq \emptyset$ if $X \overset{c}{=} Y$ for some $X \in \mathcal{S}$ and $Y \in \mathcal{T}$.

We denote by $\mathrm{perm}(n)$ the set of all permutations $p : \{0,1\}^n \to \{0,1\}^n$. Let $\mathcal{S}$ be a set. $\exists^{\neq} x, y \in S$ denotes the existence of two distinct elements in $\mathcal{S}$. Moreover, if $\mathcal{S}$ is finite, we denote by $S \xleftarrow{\$} \mathcal{S}$ the uniform random drawing of S from $\mathcal{S}$. Finally, we use the symbol E to represent the expectation.

Adversaries and Distinguishing Advantages. An adversary $\mathcal{A}$ is an algorithm that is given access to one or more oracles $\mathcal{O}$. If $\mathcal{A}$ is a distinguishing adversary, after its interaction it has to output a decision bit $b \in \{0,1\}$. We denote this as "$\mathcal{A}^{\mathcal{O}} \to b$". For two randomized oracles $\mathcal{O}$ and $\mathcal{P}$, we denote the advantage of an adversary $\mathcal{A}$ in distinguishing between them by

$$\Delta_{\mathcal{A}}(\mathcal{O}\ ;\ \mathcal{P}) = \left|\mathbf{Pr}\left(\mathcal{A}^{\mathcal{O}} \to 1\right) - \mathbf{Pr}\left(\mathcal{A}^{\mathcal{P}} \to 1\right)\right| .$$

We will only be concerned with computationally unbounded adversaries $\mathcal{A}$, whose complexities are only measured by the number and type of oracle queries. Without loss of generality, we assume that an adversary never makes queries for which it already knows the answer.

3 Ascon Mode

Let $k, m, n, c, r \in \mathbb{N}$ with $c + r = n$, $k + m \leq n$, and $m \leq k$. The authenticated encryption scheme Ascon consists of two algorithms, the encryption algorithm $\mathcal{E}$ and decryption algorithm $\mathcal{D}$. Encryption $\mathcal{E}$ takes as input a key $K \in \{0,1\}^k$, a nonce $N \in \{0,1\}^m$, associated data $A \in \{0,1\}^*$, and plaintext $P \in \{0,1\}^*$, and it outputs a ciphertext C of the exact same size as P, so $C \in \{0,1\}^{|P|}$, and a tag $T \in \{0,1\}^m$:

$$\mathcal{E}(K, N, A, P) = (C, T)\,.$$

The corresponding decryption function $\mathcal{D}$ takes as input a key $K \in \{0,1\}^k$, a nonce $N \in \{0,1\}^m$, associated data $A \in \{0,1\}^*$, and ciphertext $C \in \{0,1\}^*$, and a tag $T \in \{0,1\}^m$, and it outputs either P of the exact same size as C if the tag is correct or a failure symbol $\perp$:

$$\mathcal{D}(K, N, A, C, T) \in \{0,1\}^{|C|} \cup \{\perp\}\,.$$

In our work, we will focus on the *mode* of Ascon, which by abuse of notation we still call Ascon. The encryption and decryption algorithms of this mode are discussed in Section 3.1. In Section 3.2 we compare the mode with the actual Ascon scheme, pinpoint the differences, and argue how these differences affect implication of our security observations to the real scheme.

3.1 Encryption and Decryption

The encryption and decryption functions internally operate on top of two permutations $p, q \in \mathrm{perm}(n)$. Let $IV \in \{0,1\}^{n-k-m}$ be any fixed initialization vector. The Ascon encryption mode is depicted in Fig. 1a and its decryption mode in Fig. 1b. Here, the "Associated data" phase is only evaluated for non-empty associated data, and in that case the associated data is 10-padded. The "Plaintext"/"Ciphertext" phase is always executed, noting that the plaintext is also 10-padded. The ciphertext is truncated to the size of the plaintext.

3.2 Comparison With Ascon

The parametrization of our mode versus the actual Ascon scheme differs in various aspects. First off, the Ascon developers fix $k = m = 128$, i.e., restrict their focus on keys, nonces, and tags of size 128 bits. We opted to keep k and m general in the mode to obtain a security result that more accurately exposes the role of the key and nonce/tag size.

The actual Ascon scheme also encodes parameters in the IV, meaning that it is of size at least 32 bits. As such, they restrict the key to size at most 160 bits. As our analysis is for a fixed parameter set, this encoding of the parameters in IV does not play a role. We therefore simply assume a fixed IV and its length is irrelevant for our analysis: all we require is that $k + m \leq n$, next to $m \leq k$ in order to have the tag fully blinded by the key.

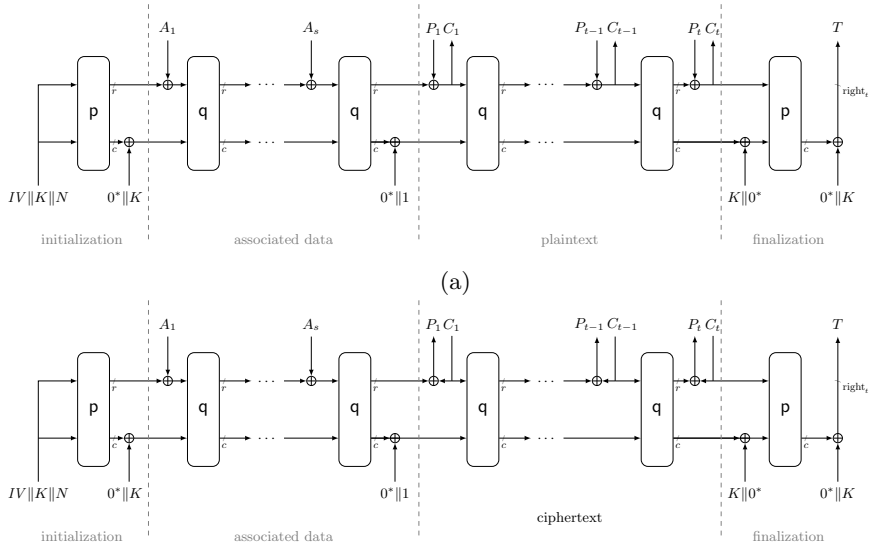

Fig. 1: The Ascon mode of operation: (a) encryption $\mathcal{E}$ and (b) decryption $\mathcal{D}$, both generalized in parameter size. In both cases, the associated data $A \in \{0,1\}^*$ is first injectively padded into r-bit blocks as $A_1\|\cdots\|A_s \leftarrow \mathrm{pad}_r^{10^*}(A)$. The plaintext is likewise padded into r-bit blocks as $P_1\|\cdots\|P_t \leftarrow \mathrm{pad}_r^{10^*}(P)$. For decryption, however, the ciphertext is padded as $C_1\|\cdots\|C_t \leftarrow \mathrm{pad}_r(C)$ and an additional 1 is added (not depicted) to cope with the injective padding upon encryption.

Apart from the parameters specified for our scheme, Ascon also introduces parameters a and b: these correspond to the number of *rounds* in the permutations p and q. In detail, in Ascon the permutation p consists of $a = 12$ rounds whereas q consists of its last $b = 6$ or $b = 8$ rounds (depending on the version). We discard the overlap and simply assume that the two permutations are independent.

We will consider security in the random permutation model where $p, q \xleftarrow{\$} \mathrm{perm}(n)$. This is, as a matter of fact, the most crucial difference between our description and that of the actual Ascon: our analysis will demonstrate only resistance against generic attacks; actual attacks on Ascon may use internal properties of p, q, and these are not captured by our security analysis.

4 Security Model

We will discuss the conventional model for authenticated encryption in Section 4.1, its separation into confidentiality and authenticity in Section 4.2, and our definition of authenticity under state recovery in Section 4.3.

4.1 Authenticated Encryption Security

We will investigate security of the Ascon mode in the random permutation model. This means that we assume random permutations $p, q \xleftarrow{\$} \mathrm{perm}(n)$. We will consider multi-user security of Ascon, where the adversary can simultaneously query up to $\mu \geq 1$ versions of the scheme. The multi-user security of Ascon against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mu\text{-ae}}_{\text{Ascon}}(\mathcal{A}) = \Delta_{\mathcal{A}}\left(\left(\mathcal{E}^{p,q}_{K_j}, \mathcal{D}^{p,q}_{K_j}\right)_{j=1}^{\mu}, p^{\pm}, q^{\pm}\ ;\ \left(\$_j, \bot\right)_{j=1}^{\mu}, p^{\pm}, q^{\pm}\right), \quad (1)$$

where $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$, $p, q \xleftarrow{\$} \mathrm{perm}(n)$, and $\$_1, \ldots, \$_\mu$ are random functions that for each new input (N, A, P) generate a random string of size $|P| + m$ bits. Here, the superscript "$\pm$" refers to bidirectional query access. The function $\bot$ returns the failure symbol $\bot$ for each query.

The adversary is not allowed to make a decryption query on input of the result of an earlier encryption query. In addition, we call $\mathcal{A}$ *nonce-respecting* if every encryption query is made for a nonce N that is different from all nonces used in earlier encryption queries under the same key. Note: in this case $\mathcal{A}$ is allowed to re-use a nonce in a decryption query or to re-use a nonce in an encryption query that was used in an earlier decryption query.

We will typically bound the adversarial complexity by $Q_{\mathcal{E}}$ encryption queries (to $(\mathcal{E}^{p}_{K_j})_{j=1}^{\mu}$ or $(\$_j)_{j=1}^{\mu}$) with a total amount of $\sigma_{\mathcal{E}}$ blocks,[2] $Q_{\mathcal{D}}$ decryption queries (to $(\mathcal{D}^{p}_{K_j})_{j=1}^{\mu}$ or $(\bot)_{j=1}^{\mu}$) with a total amount of $\sigma_{\mathcal{D}}$ blocks, Q_p primitive queries to p, p^{-1}, and Q_q primitive queries to q, q^{-1}.

4.2 Separation Into Confidentiality and Authenticity

In order to analyze authenticity both against nonce-respecting and nonce-misusing adversaries, it will be convenient for us to separate the security of Section 4.1 into confidentiality and authenticity:

$$\mathbf{Adv}^{\mu\text{-conf}}_{\text{Ascon}}(\mathcal{A}) = \Delta_{\mathcal{A}}\left(\left(\mathcal{E}^{p,q}_{K_j}\right)_{j=1}^{\mu}, p^{\pm}, q^{\pm}\ ;\ \left(\$_j\right)_{j=1}^{\mu}, p^{\pm}, q^{\pm}\right), \quad (2)$$

$$\mathbf{Adv}^{\mu\text{-auth}}_{\text{Ascon}}(\mathcal{A}) = \mathbf{Pr}\left(\mathcal{A}^{\left(\mathcal{E}^{p,q}_{K_j}, \mathcal{D}^{p,q}_{K_j}\right)_{j=1}^{\mu}, p^{\pm}, q^{\pm}} \text{ forges}\right), \quad (3)$$

[2] A "block" in this case is counted as the number of q-evaluations that would be induced in the real world. This definition seems counter-intuitive at first sight, as a single call to Ascon for empty associated data and a one-block plaintext incurs 0 q-evaluations. However, for the security proof this is the most logical definition.

where for authenticity, we say that $\mathcal{A}$ "forges" if it ever makes a query to one of its decryption oracles that is successful and that is not the result of an earlier encryption query. The same remarks on the randomness and the query complexities as in Section 4.1 apply.

It can be observed [6] that

$$\mathbf{Adv}^{\mu\text{-ae}}_{\text{Ascon}}(\mathcal{A}) \leq \mathbf{Adv}^{\mu\text{-conf}}_{\text{Ascon}}(\mathcal{A}') + \mathbf{Adv}^{\mu\text{-auth}}_{\text{Ascon}}(\mathcal{A}'')\,,$$

where $\mathcal{A}'$ and $\mathcal{A}''$ have the same query complexities as $\mathcal{A}$. This separation allows for a modular proof, but it comes at the price of some terms in the bounds being counted twice.

4.3 Authenticity Under State Recovery

One property claimed by the designers of Ascon is that, if an inner state of Ascon is leaked, mounting forgeries or recovering the keys is hard. Intuitively, the idea is that (i) whenever an inner state is leaked to the adversary, (ii) it may evaluate itself in forward/inverse direction using evaluations of $q^{\pm}$, but (iii) it cannot get across the outer permutation evaluations due to the key blinding. After all, the secret target values outside these permutation evaluations are the key and the tag; hence the claim of the designers.

Now, obviously, authenticity is a weaker property than key recovery security, noting that if one can recover the key, one can mount a forgery. Thus, in order to capture security of Ascon under state recovery, we aim for authenticity. However, formalizing this security property is not trivial.

In order to formalize the notion, we take inspiration from the work on permutation-based leakage resilience [24–27]. In leakage resilience, the attacker has, besides the oracles of (3), access to a leaky version of the scheme where the adversary not only learns the actual inputs but also leakage of each permutation call. The adversary would then win if it can forge the challenge version of the scheme [4]. We follow the same approach in our definition of authenticity under state recovery, with the fundamental differences that (i) leakage only happens for the inner permutation q and (ii) the function leaks the entire input and output state of those permutation calls. Point (i) is supported by the idea that, at least in Ascon, the outer permutations are stronger than the inner ones, and in a leveled implementation [37] one could consider an application where these outer permutation calls that are surrounded by key material are better masked. Point (ii) is mostly out of generosity: the attacker learns *all* internal secret states, rather than just a selected amount of them. This is without loss of generality as the adversary can proceed itself from any leaked secret state using $q^{\pm}$ anyway.

Formally, we define the notion of authenticity under state recovery as the definition of authenticity of (3), where $\mathcal{A}$ gets access to *learning oracles* $\mathcal{LE}$ and $\mathcal{LD}$, which are defined as $\mathcal{E}$ and $\mathcal{D}$ but that additionally leak all input/output values of the evaluations of inner permutation q (cf., Fig. 1). It wins if it ever makes a query to one of its learning decryption oracles that is successful and

that is not the result of an earlier encryption query. This leads to the following model:

$$\mathbf{Adv}^{\mu\text{-auth}}_{\text{Ascon}}(\mathcal{A}) = \mathbf{Pr}\left(\mathcal{A}^{\left(\mathcal{LE}^{p,q}_{K_j},\mathcal{LD}^{p,q}_{K_j}\right)^{\mu}_{j=1},p^{\pm},q^{\pm}} \text{ forges}\right). \tag{4}$$

We consider the setting where the adversary is allowed to repeat nonces. We remark that the definition is almost equivalent to muCIML2 of [26, Definition 2], the main differences being that they consider two structurally different underlying ideal primitives rather than one, and that we make explicit what conditions apply on the nonce.

5 Security of Ascon Mode

We prove nonce-based confidentiality of the Ascon mode in Section 5.1, authenticity under nonce misuse in Section 5.2, and nonce-based authenticity in Section 5.3. We prove authenticity of the Ascon mode in both the nonce-respecting as in the nonce-misuse setting. In both authenticity proofs, the adversary is allowed to repeat nonces under decryption queries; the sole difference in the two settings is that in the nonce-respecting setting the adversary is not allowed to use a single nonce for a single key twice for encryption. Authenticity under state recovery is analyzed in Section 5.4.

5.1 Confidentiality

We prove confidentiality of the Ascon mode in the nonce-based setting, where the adversary is not allowed to reuse nonces for different calls to a single encryption oracle. It is allowed to repeat nonces under different keys.

Theorem 1. *Let $k, m, n, c, r, \mu \in \mathbb{N}$ with $c + r = n$, $k + m \leq n$, and $m \leq k$, and consider the mode* Ascon $= (\mathcal{E}, \mathcal{D})$ *of Section 3. For any nonce-respecting adversary $\mathcal{A}$ making at most $Q_{\mathcal{E}}$ encryption queries with a total amount of $\sigma_{\mathcal{E}}$ blocks, Q_p primitive queries to p, and Q_q primitive queries to q, such that $Q_p \leq \min\{2^{k-1}, 2^{c-1}\}$ and $Q_q \leq 2^{c-1}$,*

$$\begin{aligned}\mathbf{Adv}^{\mu\text{-conf}}_{\text{Ascon}}(\mathcal{A}) \leq{}& \binom{\mu}{2}\frac{1}{2^k} + \frac{2\mu Q_p}{2^k} + \frac{2\mu Q_{\mathcal{E}}}{2^{n-m}} + \frac{(6m+8)Q_p}{2^{n-m}} \\ &+ \frac{(\sigma_{\mathcal{E}} + 2Q_{\mathcal{E}})^2}{2^n} + \frac{12Q_q\,(Q_{\mathcal{E}} + \sigma_{\mathcal{E}})}{2^n} + \frac{(2Q_{\mathcal{E}} + Q_p)^2}{2^n} + \frac{(\sigma_{\mathcal{E}} + Q_q)^2}{2^n} \\ &+ \frac{16Q_pQ_{\mathcal{E}}}{2^n} + \frac{(6r+8)Q_q}{2^c} + \frac{(12r+16)Q_p}{2^c}.\end{aligned}$$

At a high level, if M denotes the total number of encryption blocks queried, and N the number of primitive queries, this bound is of the form

$$\mathcal{O}\left(\frac{\mu^2}{2^k} + \frac{\mu(M+N)}{2^k} + \frac{(N+M)^2}{2^n} + \frac{N}{2^c}\right).$$

The proof is roughly inspired by that of Jovanovic et al. [30,31] but uses a more novel multicollision bounding technique. In addition, and more importantly, it explicitly takes the keying of the Ascon mode into account.

Proof (of Theorem 1). Let $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$ be μ random keys, $p, q \xleftarrow{\$} \mathrm{perm}(n)$ be two random permutations, and $\$_1, \ldots, \$_\mu$ be μ random functions. Let $\mathcal{A}$ be a nonce-respecting adversary. Our goal is to bound the distance $\mathbf{Adv}^{\mu\text{-conf}}_{\text{Ascon}}(\mathcal{A})$ of (2).

RP-RF Switch. As a first step, we replace the primitives $p^\pm$ and $q^\pm$ by lazily sampled random functions $f_p^\pm$ and $f_q^\pm$ by using the RP-RF switching lemma [7]. More precisely, these primitives maintain distinct lists $\mathcal{L}_p$ and $\mathcal{L}_q$, respectively, containing tuples of the form (X, Y). On a forward query X, the function checks whether X appears as first position in the list. If this is not the case, it samples $Y \xleftarrow{\$} \{0,1\}^n$, and adds (X, Y) to the corresponding list. Otherwise, it simply replies with any Y such that (X, Y) is in the list. The functions operate comparably for inverse queries. As long as no collisions occur in these lists, then $(f_p^\pm, f_q^\pm)$ are indistinguishable from $(p^\pm, q^\pm)$. Moreover, there are $2Q_\mathcal{E} + Q_p$ queries to the primitive f_p and $\sigma_\mathcal{E} + Q_q$ queries to the primitive f_q, therefore

$$\mathbf{Adv}^{\mu\text{-conf}}_{\text{Ascon}}(\mathcal{A}) \leq \Delta_\mathcal{A}\left(\left(\mathcal{E}_{K_j}^{f_p,f_q}\right)_{j=1}^{\mu}, f_p^\pm, f_q^\pm \;;\; (\$_j)_{j=1}^{\mu}, f_p^\pm, f_q^\pm\right) + \frac{(2Q_\mathcal{E} + Q_p)^2}{2^n} + \frac{(\sigma_\mathcal{E} + Q_q)^2}{2^n}\,. \quad (5)$$

Transcripts. For ease of notation, we will denote the real world as $W_R = \left(\left(\mathcal{E}_{K_j}^{f_p,f_q}\right)_{j=1}^{\mu}, f_p^\pm, f_q^\pm\right)$ and the ideal world as $W_I = \left((\$_j)_{j=1}^{\mu}, f_p^\pm, f_q^\pm\right)$. We define a transcript τ for the real world. This transcript records all evaluations of $f_p^\pm$ and $f_q^\pm$, either through construction or primitive queries. In detail, τ contains tuples of the form $(X, Y, \mathrm{d}, \mathcal{O}, j)$:

- X denotes the input, Y, the output, and $\mathrm{d} \in \{\mathrm{fwd}, \mathrm{inv}\}$ the query direction;
- $\mathcal{O} \in \{p, q, \mathcal{E}_p^{\text{beg}}, \mathcal{E}_p^{\text{end}}, \mathcal{E}_q\}$ indicates whether the evaluation is a f_p-query from the adversary, a f_q-query from the adversary, a first encryption f_p-evaluation, a last encryption f_p-evaluation, or an encryption f_q-evaluation, respectively. We abuse notation and denote by $\mathcal{O} = \mathcal{E}_p$ that $\mathcal{O} \in \{\mathcal{E}_p^{\text{beg}}, \mathcal{E}_p^{\text{end}}\}$;
- $j \in [\![0, \mu]\!]$ indicates the corresponding key index in construction queries, or 0 if $\mathcal{O} \in \{p, q\}$.

Bad Events. We now define bad event **BAD** over the extended transcript, which is split into two bad events: $\mathbf{BAD} := \mathbf{GUESS} \vee \mathbf{COL}$.

Bad event **GUESS** itself is further split into $\mathbf{GUESS} := \mathbf{GUESS}^{\text{key}} \vee \mathbf{GUESS}_p \vee \mathbf{GUESS}_q$. These bad events are defined as follows:

$$\begin{aligned}
\mathbf{GUESS}^{\text{key}} &: \exists (X, Y, \mathrm{d}, p, 0) \in \tau, j \in [\![1, \mu]\!] \text{ such that } \lceil X \rceil_{n-m} = IV \parallel K_j\,,\\
\mathbf{GUESS}_p &: \exists (X, Y, \mathrm{d}, p, 0), (U, V, \mathrm{fwd}, \mathcal{E}_p, j) \in \tau \text{ such that } X = U\,,\\
\mathbf{GUESS}_q &: \exists (X, Y, \mathrm{d}, q, 0), (U, V, \mathrm{fwd}, \mathcal{E}_q, j) \in \tau \text{ such that } X = U\,.
\end{aligned}$$

The purpose of **GUESS** is to capture the case that the adversary "guesses" an intermediate state that was generated during a construction query. Event $\mathbf{GUESS}^{\text{key}}$ corresponds to guessing the key, while for the other bad events, the subscript indicates if this guess is a guess for the outer primitive p or the inner primitive q.

Bad event **COL** itself is further split into $\mathbf{COL} := \mathbf{COL}^{\text{key}} \vee \mathbf{COL}^{\text{aux}} \vee \mathbf{COL}^{\text{st}}$. $\mathbf{COL}^{\text{st}}$ is further split into $\mathbf{COL}^{\text{st}} := \mathbf{COL}^{\text{st}}_{\text{inter}} \vee \mathbf{COL}^{\text{st}}_{\text{final}}$. These bad events are defined as follows:

$$\mathbf{COL}^{\text{key}} : \exists^{\neq} j, j' \in [\![1, \mu]\!] \text{ such that } K_j = K_{j'} ,$$
$$\mathbf{COL}^{\text{aux}} : \exists (U, V, \text{fwd}, \mathcal{E}_p^{\text{end}}, j), j' \in [\![1, \mu]\!] \text{ such that } \lceil U \rceil_{n-m} = IV \parallel K_{j'} ,$$
$$\mathbf{COL}^{\text{st}}_{\text{inter}} : \exists^{\neq} (U, V, \text{fwd}, \mathcal{E}_q, j), (U', V', \text{fwd}, \mathcal{E}_q, j') \in \tau \text{ such that } U = U' ,$$
$$\mathbf{COL}^{\text{st}}_{\text{final}} : \exists^{\neq} (U, V, \text{fwd}, \mathcal{E}_p^{\text{end}}, j), (U', V', \text{fwd}, \mathcal{E}_p^{\text{end}}, j') \in \tau \text{ such that } U = U' .$$

The purpose of **COL** is to handle collisions between the keys or between intermediate states in constructions queries. It thus guarantees that all permutation queries made by construction calls in the real world are fresh (provided, of course, no such state is "guessed" in a primitive query). In detail, $\mathbf{COL}^{\text{key}}$ ensures that no collisions between two initial states $IV \parallel K_j \parallel N$ and $IV \parallel K_{j'} \parallel N'$ occur, $\mathbf{COL}^{\text{aux}}$ prevents collisions between an initial state and a state before the last f_p-evaluation, and $\mathbf{COL}^{\text{st}}$ handles the remaining collisions.

Further Steps. In Lemma 1, we show that, as long as **BAD** does not occur in the real world W_R, the worlds W_R and W_I are indistinguishable. Then, in Lemma 2 we upper bound the probability that **BAD** occurs in the real world W_R. Together with (5), these two lemmas complete the proof. □

Lemma 1. *As long as* **BAD** *does not occur in* W_R*, worlds* W_R *and* W_I *are indistinguishable. Formally,*

$$\Delta_{\mathcal{A}} (W_R \,;\, W_I) \leq \mathbf{Pr} \left(\mathcal{A}^{W_R} \textit{ sets } \mathbf{BAD} \right) .$$

Proof. As long as **GUESS** does not occur in W_R, the adversarial primitive queries do not coincide with any of the primitive evaluations made in construction queries. This means that we can restrict our focus to the difference between the output distributions of the worlds $W'_R = \left(\mathcal{E}_{K_j}^{f_p, f_q}\right)_{j=1}^{\mu}$ and $W'_I = \left(\$_j\right)_{j=1}^{\mu}$. Note that, as each query is made for a new nonce, the outputs in W'_I are uniformly random string (of length $|P| + m$ bits, where $|P|$ is the length of the plaintext of the corresponding query). For W'_R, events $\mathbf{COL}^{\text{key}}$ and $\mathbf{COL}^{\text{aux}}$ make sure that each construction evaluation starts with a new initial input to f_p, event $\mathbf{COL}^{\text{st}}_{\text{inter}}$ assures that all internal calls to f_q are distinct, and event $\mathbf{COL}^{\text{st}}_{\text{final}}$ assures that all final calls to f_p are distinct. As f_p and f_q are random functions, this means that the ciphertexts and tags in W'_R are uniform random strings, just like in W'_I.

We thus obtain that, as long as **BAD** does not occur in W_R, in both worlds the outputs will always be uniformly randomly generated strings. This would complete the assertion by the fundamental lemma of game playing [7]. □

Lemma 2. *We have*

$$\begin{aligned}\mathbf{Pr}\left(\mathcal{A}^{W_R} \text{ sets } \mathbf{BAD}\right) \le \binom{\mu}{2}\frac{1}{2^k} + \frac{2\mu Q_p}{2^k} + \frac{2\mu Q_{\mathcal{E}}}{2^{n-m}} + \frac{(6m+8)Q_p}{2^{n-m}} \\ + \frac{(\sigma_{\mathcal{E}} + 2Q_{\mathcal{E}})^2}{2^n} + \frac{12Q_q\left(Q_{\mathcal{E}} + \sigma_{\mathcal{E}}\right)}{2^n} \\ + \frac{16Q_pQ_{\mathcal{E}}}{2^n} + \frac{(6r+8)Q_q}{2^c} + \frac{(12r+16)Q_p}{2^c}\,.\end{aligned}$$

The proof can be found in Appendix B.

5.2 Authenticity in Nonce-Misuse Setting

We prove the following result for authenticity against possibly nonce-misusing adversaries.

Theorem 2. *Let $k, m, n, c, r, \mu \in \mathbb{N}$ with $c + r = n$, $k + m \le n$, and $m \le k$, and consider the mode* $\mathrm{Ascon} = (\mathcal{E}, \mathcal{D})$ *of Section 3. For any possibly nonce-misusing adversary $\mathcal{A}$ making at most $Q_{\mathcal{E}}$ encryption queries with a total amount of $\sigma_{\mathcal{E}}$ blocks, $Q_{\mathcal{D}}$ decryption queries with a total amount of $\sigma_{\mathcal{D}}$ blocks, Q_p primitive queries to p, and Q_q primitive queries to q, such that $2Q_{\mathcal{E}} + 2Q_{\mathcal{D}} + Q_p \le 2^{c-1}$ and $\sigma_{\mathcal{E}} + \sigma_{\mathcal{D}} + Q_p \le 2^{c-1}$,*

$$\begin{aligned}\mathbf{Adv}^{\mu\text{-auth}}_{\mathrm{Ascon}}(\mathcal{A}) \le \frac{2Q_{\mathcal{D}}}{2^m} + \binom{\mu}{2}\frac{1}{2^k} + \frac{2\mu(Q_{\mathcal{E}} + Q_{\mathcal{D}} + Q_p)}{2^k} \\ + \frac{Q_p(6m+8)}{2^{n-m}} + \frac{4Q_p(Q_{\mathcal{E}} + Q_{\mathcal{D}})}{2^n} \\ + \frac{16(Q_p + Q_q + Q_{\mathcal{E}} + Q_{\mathcal{D}} + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}})(Q_{\mathcal{E}} + Q_{\mathcal{D}} + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}})}{2^c}\,.\end{aligned}$$

At a high level, if M represents the total number of encryption and decryption blocks queried, and N the number of primitive queries, this bound is of the form

$$\mathcal{O}\left(\frac{2Q_{\mathcal{D}}}{2^m} + \frac{\mu^2}{2^k} + \frac{\mu(M+N)}{2^k} + \frac{M(N+M)}{2^c} + \frac{N}{2^c}\right).$$

The proof relies in part on that of confidentiality (Theorem 1) but differs in that the adversary is allowed to reuse nonces and henceforth may potentially set outer parts of states to a value of its choice. The nonce reuse leads to an additional problem, namely that an intermediate state for some construction evaluation could become a final state for another one. These two issues are covered by updating the bad events as required.

Proof (of Theorem 2). Let $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$ be μ random keys, and $p, q \xleftarrow{\$} \mathrm{perm}(n)$ be two random permutations. Let $\mathcal{A}$ be a possibly nonce-misusing adversary. Our goal is to bound the probability $\mathbf{Adv}^{\mu\text{-auth}}_{\mathrm{Ascon}}(\mathcal{A})$ of (3).

Setup. Unlike the proof of confidentiality (Theorem 1), we do not aim for a distinguishing event, and we will also not resort to an RP-RF-switch. However, we adopt the extended transcript notation and expand it to decryption queries. In detail, the transcript τ records all evaluations of the form $(X, Y, \mathrm{d}, \mathcal{O}, j)$, where now the origin satisfies $\mathcal{O} \in \{p, q, \mathcal{E}_p^{\mathrm{beg}}, \mathcal{E}_p^{\mathrm{end}}, \mathcal{E}_q, \mathcal{D}_p^{\mathrm{beg}}, \mathcal{D}_p^{\mathrm{end}}, \mathcal{D}_q\}$, in order to take into account decryption queries. Abusing notation, we write $\mathcal{C}$ to denote $\mathcal{E}$ or $\mathcal{D}$.

Looking ahead, we will define a bad event **M-BAD**. We have

$$\mathbf{Pr}\left(\mathcal{A}^{\left(\mathcal{E}_{K_j}^{p,q}, \mathcal{D}_{K_j}^{p,q}\right)_{j=1}^{\mu}, p^{\pm}, q^{\pm}} \text{ forges}\right) \leq \mathbf{Pr}\left(\mathcal{A}^{\left(\mathcal{E}_{K_j}^{p,q}, \mathcal{D}_{K_j}^{p,q}\right)_{j=1}^{\mu}, p^{\pm}, q^{\pm}} \text{ forges} \;\middle|\; \neg\textbf{M-BAD}\right) + \mathbf{Pr}\left(\textbf{M-BAD}\right). \quad (6)$$

The rest of the proof consists of defining the bad events. The first probability in (6) is bounded in Lemma 3, and the second probability in Lemma 4. Together, these results complete the proof.

Bad Events. We will slightly change the two bad events **GUESS** and **COL** of Theorem 1 into **M-GUESS** and **M-COL**, and define **M-BAD** := **M-GUESS**$\vee$ **M-COL**.

Before defining the bad events, we first define the following sets for any construction-originated query $(X, Y, \mathrm{fwd}, \mathcal{O}, j)$:

$$\texttt{InterSt}((U, V, \mathrm{fwd}, \mathcal{O}, j)) = \begin{cases} \left\{V \oplus (0^{n-k} \parallel K_j), V \oplus (0^{n-k} \parallel K_j) \oplus 0^*1\right\}, & \text{if } \mathcal{O} = \mathcal{C}_p^{\mathrm{beg}}, \\ \{V \oplus 0^*1, V\}, & \text{if } \mathcal{O} = \mathcal{C}_q, \\ \emptyset, & \text{otherwise}, \end{cases}$$

$$\texttt{LastSt}((U, V, \mathrm{fwd}, \mathcal{O}, j)) = \begin{cases} \left\{V \oplus (0^{n-k} \parallel K_j) \oplus 0^*1 \oplus (0^r \parallel K_j \parallel 0^{c-k})\right\}, & \text{if } \mathcal{O} = \mathcal{C}_p^{\mathrm{beg}}, \\ \left\{V \oplus 0^*1 \oplus (0^r \parallel K_j \parallel 0^{c-k}), V \oplus (0^r \parallel K_j \parallel 0^{c-k})\right\}, & \text{if } \mathcal{O} = \mathcal{C}_q, \\ \emptyset, & \text{otherwise}. \end{cases}$$

The goal of these sets is based on the following observation. As nonces can repeat, it is possible that an intermediate state once generated using an encryption query could later become a final state of another query, or if this state was during absorption of associated data, then domain separator bits may be added. These sets encompass all future possibilities for the states generated by construction queries. The sets allow us to drastically simplify the bad event by excluding multiple cases at once. We will refer to elements in these sets later as *candidate states*.

We are now ready to define the bad event **M-BAD**. As in the confidentiality proof, **M-BAD** is split as **M-BAD** := **M-GUESS** $\vee$ **M-COL**.

M-GUESS itself is further split into **M-GUESS** := **M-GUESS**$^{\mathrm{key}}\vee$**M-GUESS**$_p\vee$ **M-GUESS**$_q$, where we additionally split the latter two into **M-GUESS**$_p$:=

$\mathbf{M\text{-}GUESS}_p^{\mathrm{in}} \vee \mathbf{M\text{-}GUESS}_p^{\mathrm{out}}$ and $\mathbf{M\text{-}GUESS}_q := \mathbf{M\text{-}GUESS}_q^{\mathrm{in}} \vee \mathbf{M\text{-}GUESS}_q^{\mathrm{out}}$:

$$
\begin{aligned}
\mathbf{M\text{-}GUESS}^{\mathrm{key}} &: \mathbf{GUESS}^{\mathrm{key}} \text{ (of Theorem 1)}\,,\\
\mathbf{M\text{-}GUESS}_p^{\mathrm{in}} &: \exists (X,Y,\mathrm{d},p,0),(U,V,\mathrm{fwd},\mathcal{C}_p,j) \in \tau \text{ such that } X \stackrel{c}{=} U \text{ or}\\
&\quad \exists (X,Y,\mathrm{d},p,0),(U,V,\mathrm{fwd},\mathcal{O},j) \in \tau\\
&\qquad\qquad \text{such that } X \stackrel{c}{\in} \mathtt{LastSt}((U,V,\mathrm{fwd},\mathcal{O},j))\,,\\
\mathbf{M\text{-}GUESS}_p^{\mathrm{out}} &: \exists (X,Y,\mathrm{d},p,0),(U,V,\mathrm{fwd},\mathcal{C}_p,j) \in \tau \text{ such that } Y \stackrel{c}{=} V\,,\\
\mathbf{M\text{-}GUESS}_q^{\mathrm{in}} &: \exists (X,Y,\mathrm{d},q,0),(U,V,\mathrm{fwd},\mathcal{C}_q,j) \in \tau \text{ such that } X \stackrel{c}{=} U \text{ or}\\
&\quad \exists (X,Y,\mathrm{d},q,0),(U,V,\mathrm{fwd},\mathcal{O},j) \in \tau\\
&\qquad\qquad \text{such that } X \stackrel{c}{\in} \mathtt{InterSt}((U,V,\mathrm{fwd},\mathcal{O},j))\,,\\
\mathbf{M\text{-}GUESS}_q^{\mathrm{out}} &: \exists (X,Y,\mathrm{d},q,0),(U,V,\mathrm{fwd},\mathcal{C}_q,j) \text{ such that } Y \stackrel{c}{=} V\,.
\end{aligned}
$$

Likewise, $\mathbf{M\text{-}COL}$ is split into $\mathbf{M\text{-}COL} := \mathbf{M\text{-}COL}^{\mathrm{key}} \vee \mathbf{M\text{-}COL}^{\mathrm{aux}} \vee \mathbf{M\text{-}COL}^{\mathrm{st}}$, where $\mathbf{M\text{-}COL}^{\mathrm{st}}$ is further split into $\mathbf{M\text{-}COL}^{\mathrm{st}} = \mathbf{M\text{-}COL}^{\mathrm{st}}_{\mathrm{inter}} \vee \mathbf{M\text{-}COL}^{\mathrm{st}}_{\mathrm{final}}$:

$$
\begin{aligned}
\mathbf{M\text{-}COL}^{\mathrm{key}} &: \mathbf{COL}^{\mathrm{key}} \text{ (of Theorem 1)}\,,\\
\mathbf{M\text{-}COL}^{\mathrm{aux}} &: \mathbf{COL}^{\mathrm{aux}} \text{ (of Theorem 1)}\,,\\
\mathbf{M\text{-}COL}^{\mathrm{st}}_{\mathrm{inter}} &: \exists^{\neq} (U,V,\mathrm{fwd},\mathcal{O},j),(U',V',\mathrm{fwd},\mathcal{O}',j') \in \tau \text{ such that}\\
&\quad \mathtt{InterSt}((U,V,\mathrm{fwd},\mathcal{O},j)) \stackrel{c}{\cap} \mathtt{InterSt}((U',V',\mathrm{fwd},\mathcal{O}',j')) \neq \emptyset\,,\\
\mathbf{M\text{-}COL}^{\mathrm{st}}_{\mathrm{final}} &: \exists^{\neq} (U,V,\mathrm{fwd},\mathcal{O},j),(U',V',\mathrm{fwd},\mathcal{O}',j') \in \tau \text{ such that}\\
&\quad \mathtt{LastSt}((U,V,\mathrm{fwd},\mathcal{O},j)) \stackrel{c}{\cap} \mathtt{LastSt}((U',V',\mathrm{fwd},\mathcal{O}',j')) \neq \emptyset\,.
\end{aligned}
$$

At a high level, the roles of $\mathbf{M\text{-}COL}^{\mathrm{st}}_{\mathrm{inter}}$ and $\mathbf{M\text{-}COL}^{\mathrm{st}}_{\mathrm{final}}$ are similar to that of $\mathbf{COL}^{\mathrm{st}}$, but with the equalities replaced by equalities on the inner part. Unlike the proof of confidentiality, the nonces can repeat, meaning that an intermediate state can appear with a different form in a later query. Consequently, we have modified the bad events $\mathbf{M\text{-}COL}^{\mathrm{st}}$ to encompass all future possibilities for the states. While it could be argued that capturing all possibilities might be unnecessary, the definitions will be crucial in the context of a nonce-respecting adversary (Theorem 3). Indeed, in this case we want to ensure that if a state generated during a decryption query triggers **BAD** when used in a subsequent encryption query, then **BAD** must have already been triggered during the original decryption query. □

Lemma 3. *We have*

$$
\mathbf{Pr}\left(\mathcal{A}^{\left(\mathcal{E}_{K_j}^{p,q},\mathcal{D}_{K_j}^{p,q}\right)_{j=1}^{\mu},p^{\pm},q^{\pm}} \textit{ forges} \;\middle|\; \neg\mathbf{M\text{-}BAD}\right) \leq \frac{2Q_{\mathcal{D}}}{2^m}\,.
$$

Proof. Conditioned on $\neg\mathbf{M\text{-}BAD}$, the state from which the tag is extracted is freshly generated during the decryption query. It is therefore sampled uniformly

at random from a set of size at least $2^n - Q_p - 2Q_{\mathcal{E}} - 2Q_{\mathcal{D}}$, among which at most 2^{n-m} values give a successful forgery. Therefore, that forgery succeeds with probability at most $2^{n-m}/2^n - Q_p - 2Q_{\mathcal{E}} - 2Q_{\mathcal{D}} \leq 2/2^m$, using that $Q_p + 2Q_{\mathcal{E}} + 2Q_{\mathcal{D}} \leq 2^{n-1}$. The result is obtained by summing over all $Q_{\mathcal{D}}$ decryption queries. □

Lemma 4. *We have*

$$\begin{aligned}\mathbf{Pr}\left(\textbf{M-BAD}\right) \leq \binom{\mu}{2}\frac{1}{2^k} + \frac{2\mu(Q_{\mathcal{E}}+Q_{\mathcal{D}}+Q_p)}{2^k} \\ + \frac{Q_p(6m+8)}{2^{n-m}} + \frac{4Q_p(Q_{\mathcal{E}}+Q_{\mathcal{D}})}{2^n} \\ + \frac{16(Q_p+Q_q+Q_{\mathcal{E}}+Q_{\mathcal{D}}+\sigma_{\mathcal{E}}+\sigma_{\mathcal{D}})(Q_{\mathcal{E}}+Q_{\mathcal{D}}+\sigma_{\mathcal{E}}+\sigma_{\mathcal{D}})}{2^c}\,.\end{aligned}$$

The proof is available in the full version of this paper [32]. Nonetheless, it closely resembles that of Lemma 2, and the main difference lies in the evaluation of the bad events on their inner part only.

5.3 Authenticity in Nonce-Respecting Setting

We prove the following result for authenticity against nonce-respecting adversaries.

Theorem 3. *Let $k, m, n, c, r, \mu \in \mathbb{N}$ with $c + r = n$, $k + m \leq n$, and $m \leq k$, and consider the mode* Ascon $= (\mathcal{E}, \mathcal{D})$ *of Section 3. For any nonce-respecting adversary $\mathcal{A}$ making at most $Q_{\mathcal{E}}$ encryption queries with a total amount of $\sigma_{\mathcal{E}}$ blocks, $Q_{\mathcal{D}}$ decryption queries with a total amount of $\sigma_{\mathcal{D}}$ blocks, Q_p primitive queries to p, and Q_q primitive queries to q, such that $2Q_{\mathcal{E}}+Q_p \leq 2^{n-1}$, $\sigma_{\mathcal{E}}+Q_q \leq 2^{n-1}$, $Q_p + 2Q_{\mathcal{D}} \leq 2^{c-1}$, and $Q_q + \sigma_{\mathcal{D}} \leq 2^{c-1}$,*

$$\begin{aligned}\mathbf{Adv}^{\mu\text{-auth}}_{\text{Ascon}}(\mathcal{A}) \leq \frac{2Q_{\mathcal{D}}}{2^m} + \binom{\mu}{2}\frac{1}{2^k} + \frac{2\mu(Q_{\mathcal{E}}+Q_{\mathcal{D}}+Q_p)}{2^k} + \frac{Q_p(12m+16)}{2^{n-m}} \\ + \frac{4Q_pQ_{\mathcal{D}}}{2^n} + \frac{(2Q_{\mathcal{E}}+\sigma_{\mathcal{E}})^2}{2^n} + \frac{Q_pQ_{\mathcal{E}}}{2^n} + \frac{10Q_q(Q_{\mathcal{E}}+\sigma_{\mathcal{E}})}{2^n} \\ + \frac{Q_p(12r+6)}{2^c} + \frac{Q_q(6r+8)}{2^c} + \frac{12(Q_p+Q_q)(Q_{\mathcal{D}}+\sigma_{\mathcal{D}})}{2^c} \\ + \frac{8(2Q_{\mathcal{D}}+\sigma_{\mathcal{D}})(4Q_{\mathcal{E}}+2\sigma_{\mathcal{E}}+2Q_{\mathcal{D}}+\sigma_{\mathcal{D}})}{2^c}\,.\end{aligned}$$

At a high level, if $M_{\mathcal{E}}$ and $M_{\mathcal{D}}$ denote respectively the total number of encryption and decryption blocks queried, and N the number of primitive queries, this bound is of the form

$$\mathcal{O}\left(\frac{2Q_{\mathcal{D}}}{2^m} + \frac{\mu^2}{2^k} + \frac{\mu(M+N)}{2^k} + \frac{M_{\mathcal{E}}(N+M_{\mathcal{E}})}{2^n} + \frac{M_{\mathcal{D}}(N+M_{\mathcal{E}}+M_{\mathcal{D}})}{2^c} + \frac{N}{2^c}\right),$$

where $M := M_{\mathcal{E}} + M_{\mathcal{D}}$. The proof can be found in the full version of this paper [32]. It is very similar to that of the nonce-misuse setting (Theorem 2), with the difference that encryption queries are handled separately due to the adversary being more restricted in these.

5.4 Authenticity Under State Recovery

We will consider Ascon under the novel model of authenticity under state recovery, in Section 5.4.2. The bound is worse than previous bounds due to the power that the attacker has in current attack model, and in Section 5.4.3 we will elaborate on the tightness of our bound. However, before doing so, we will first in Section 5.4.1 consider a bad version of Ascon, called BadAsscon, that is equal to the construction of Fig. 1 but *omits all key additions except the first one*, and demonstrate that this construction fails to achieve authenticity under state recovery.

5.4.1 BadAsscon. We first demonstrate that BadAsscon admits a trivial authenticity attack under state recovery.

Proposition 1. *Let $k, m, n, c, r, \mu \in \mathbb{N}$ with $c + r = n$, $k + m \leq n$, and $m \leq k$, and consider the mode* BadAsscon $= (\mathcal{E}, \mathcal{D})$ *that equals* Ascon *of Section 3 but with all key additions except the first one omitted. There exists an adversary $\mathcal{A}$ making $Q_{\mathcal{E}} = 1$ encryption query with a total amount of $\sigma_{\mathcal{E}} = 0$ blocks, $Q_{\mathcal{D}} = 1$ decryption query with a total amount of $\sigma_{\mathcal{D}} = 0$ blocks, $Q_p = 0$ primitive queries to p, and $Q_q = 3$ primitive queries to q,*

$$\mathbf{Adv}^{\mu\text{-sr-auth}}_{\text{BadAsscon}}(\mathcal{A}) = 1\,.$$

The idea of the attack is very simple: from a single learning query one can recover the key, and once the key is recovered a forgery can be mounted. We include the attack for completeness.

Proof (of Proposition 1). We consider an adversary that first recovers the key K_1 and uses it to forge a tag under this key. The adversary operates as follows:

- It first makes any encryption learning query with empty associated data and with a plaintext of 1 block (w.l.o.g., already padded): $\mathcal{LE}^{p,q}_{K_1}(N, P)$. It learns (C, T) as well as the state S right after absorption of P and before application of permutation p;
- It queries $p^{-1}(S \oplus (P \parallel 0^{c-1} \parallel 1))$ to obtain $IV \parallel K_1 \parallel N$ and extracts K_1 from it;
- It selects a different tuple (N', P'), computes $\mathcal{E}^{p,q}_{K_1}(N, P) = (C, T)$ offline with two calls to p;
- It outputs forgery (N, C, T).

The forgery obviously succeeds with probability 1. □

5.4.2 Ascon. Now, we are ready to prove security of Ascon under state recovery.

Theorem 4. *Let $k, m, n, c, r, \mu \in \mathbb{N}$ with $c + r = n$, $k + m \leq n$, and $m \leq k$, and consider the mode* Ascon $= (\mathcal{E}, \mathcal{D})$ *of Section 3. For any possibly nonce-misusing adversary $\mathcal{A}$ making at most $Q_{\mathcal{E}}$ encryption queries with a total amount of $\sigma_{\mathcal{E}}$*

blocks, $Q_{\mathcal{D}}$ decryption queries with a total amount of $\sigma_{\mathcal{D}}$ blocks, Q_p primitive queries to p, and Q_q primitive queries to q such that $2Q_{\mathcal{E}} + 2Q_{\mathcal{D}} + Q_p \leq 2^{c-1}$, $\sigma_{\mathcal{E}} + \sigma_{\mathcal{D}} + Q_p \leq 2^{c-1}$, $Q_p \leq 2^{k-1}$,

$$\begin{aligned}\mathbf{Adv}^{\mu\text{-sr-auth}}_{\text{Ascon}}(\mathcal{A}) \leq{}& \frac{2Q_{\mathcal{D}}}{2^m} + \binom{\mu}{2}\frac{1}{2^k} + \frac{2\mu(Q_p + Q_{\mathcal{E}} + Q_{\mathcal{D}})}{2^k} + \frac{(12(c-k)+16)Q_p}{2^k} \\ &+ \frac{Q_p(6m+8)}{2^{n-m}} + \frac{4Q_p(Q_{\mathcal{E}} + Q_{\mathcal{D}})}{2^n} + \frac{8(2Q_{\mathcal{E}} + 2Q_{\mathcal{D}} + 8Q_q + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}})^2}{2^c} \\ &+ \frac{12Q_p(8Q_{\mathcal{E}} + 8Q_{\mathcal{D}} + Q_q + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}})}{2^c}\,.\end{aligned}$$

At a high level, if M represents the total number of encryption and decryption blocks queried, and N the number of primitive queries, this bound is of the form

$$\mathcal{O}\left(\frac{2Q_{\mathcal{D}}}{2^m} + \frac{\mu^2}{2^k} + \frac{\mu(M+N)}{2^k} + \frac{(N+M)^2}{2^c}\right).$$

The core idea why Ascon achieves authenticity under state recovery whereas BadAsscon of Section 5.4.1 did not is that the calls to the outer permutation p are masked by K at both sides. This means that, even if the adversary learns all intermediate states, it cannot "pass through" the permutations p. That said, the security setting also gives rise to other potential attacks, most notably as in case of state leakage the inner portion of Ascon behaves as a mere hash function of which the adversary knows all states, and for which it could potentially find inner collisions. This complicates the analysis of Theorem 4.

Before proceeding with the proof of Theorem 4, we wish to remark that Guo et al. already gave an argument of the security of Ascon in this setting in [26, Theorem 4]. However, their proof is informal (as explicitly stated by the authors) and only stated in big-O terms. A more detailed look at their informal reasoning reveals various unclear and unconvincing steps. Most importantly, the core step of their reasoning is to replace KDF_{K_i} (basically, in our terminology, the first permutation call including the key blindings) and TGF_{K_i} (the last permutation call including the key blindings) by secret random functions. This step is made at the cost of the PRF security of multi-instance partial-key Even-Mansour cipher, which the authors claim to be (in our terminology) $O\left(\frac{(Q_p+Q_q+\sigma_{\mathcal{E}}+\sigma_{\mathcal{D}})^2}{2^c}\right)$. As $Q_p + Q_q + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}}$ is de facto the total complexity, this bound seems lossy when purely focusing on the initialization and finalization. Of course, this is a minor issue as a tighter term for the multi-instance partial-key Even-Mansour cipher was already derived by Andreeva et al. [1, Theorem 4] and as the claimed term also appears in the security of the hash portion between key derivation and tag generation. What is more worrisome is that the reasoning is incorrect. To be precise: the security of the multi-instance partial-key Even-Mansour cipher is relative to the key size and not to the capacity. In Ascon, the key is of size $k \leq c$, and the bound of Andreeva et al. [1, Theorem 4] would carry over with c replaced by k to match our context. This substitution leads to a term of the order $\frac{(Q_p+Q_q)\cdot\text{multiplicity}+(Q_{\mathcal{E}}+Q_{\mathcal{D}})^2}{2^k}$. We stress that this gives a dramatic security loss,

noting that in Ascon, $k = 128$ and $c = 256$, and additionally, the multiplicity term of the partial-key Even-Mansour cipher replacing KDF_{K_i} can be as large as multiplicity $\approx \min\{Q_{\mathcal{E}} + Q_{\mathcal{D}}, 2^m\}$. As a matter of fact, in an earlier proof attempt, we independently followed the same reasoning as Guo et al., but quickly departed from it as the security loss described here turned out to be unavoidable.

Proof (of Theorem 4). Let $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$ be μ random keys, $p, q \xleftarrow{\$} \mathrm{perm}(n)$ be two random permutations, and $\$_1, \ldots, \$_\mu$ be μ random functions. Our goal is to bound the distance $\mathbf{Adv}^{\mu\text{-sr-auth}}_{\mathrm{Ascon}}(\mathcal{A})$ of (4).

Setup. We again adopt the transcript notation from the proof of nonce-misuse authenticity (Theorem 2), with one adjustment. Note that in the current security setting, the adversary can make q-queries starting from an intermediate state that was disclosed during an earlier $\mathcal{LC}$-query. In order to simplify the notation and bad event analysis, any such direct q-query will be stored as an encryption query. More precisely, if the adversary makes a forward q-query such that the input X collides on its inner part with any intermediate state U (possibly after xoring the domain separator bits), itself associated with key index j, then this adversarial q-query will be appended to τ as $(X, Y, \mathrm{fwd}, \mathcal{E}_q, j)$. Looking ahead, our bad events will prevent that inverse q-queries connect to any intermediate state, and guarantee that any q-query departs from only one intermediate state.

Looking ahead, we will define a bad event **SR-BAD**. We have

$$\mathbf{Pr}\left(\mathcal{A}^{\left(\mathcal{E}^{p,q}_{K_j}, \mathcal{D}^{p,q}_{K_j}\right)^{\mu}_{j=1}, p^{\pm}, q^{\pm}} \text{ forges}\right) \leq$$
$$\mathbf{Pr}\left(\mathcal{A}^{\left(\mathcal{E}^{p,q}_{K_j}, \mathcal{D}^{p,q}_{K_j}\right)^{\mu}_{j=1}, p^{\pm}, q^{\pm}} \text{ forges} \;\middle|\; \neg\textbf{SR-BAD}\right) + \mathbf{Pr}\left(\textbf{SR-BAD}\right). \quad (7)$$

The rest of the proof consists of defining the bad events. The first probability is bounded as

$$\mathbf{Pr}\left(\mathcal{A}^{\left(\mathcal{E}^{p,q}_{K_j}, \mathcal{D}^{p,q}_{K_j}\right)^{\mu}_{j=1}, p^{\pm}, q^{\pm}} \text{ forges} \;\middle|\; \neg\textbf{SR-BAD}\right) \leq \frac{2Q_{\mathcal{D}}}{2^m},$$

exactly as in Lemma 3 (using that $Q_p + 2Q_{\mathcal{E}} + 2Q_{\mathcal{D}} \leq 2^{n-1}$), but with a side remark that we use the fact that as long as no collisions occur between inner states, and the adversary did not guess the key or a construction evaluation of p, the last p-call in any forgery attempt is new. The second probability is bounded in Lemma 5. Together, these results complete the proof.

Bad Events. The bad events of the nonce-misuse authenticity (Theorem 2) are inherited, but we define an additional bad event called **SR-INNER**. This event captures the case where an inverse q-query connects with a candidate intermediate state. The remaining inner collisions that can be caused by adversarial queries are covered by the event $\textbf{SR-COL}^{\mathrm{st}}$.

In detail, we have $\textbf{SR-BAD} := \textbf{SR-GUESS} \vee \textbf{SR-COL} \vee \textbf{SR-INNER}$. Event **SR-GUESS** itself is further split into $\textbf{SR-GUESS} := \textbf{SR-GUESS}^{\mathrm{key}} \vee$

SR-GUESS$_p$. Event **SR-COL** is split into **SR-COL** := **SR-COL**$^{\text{key}} \vee$ **SR-COL**$^{\text{aux}} \vee$ **SR-COL**$^{\text{st}}$. The individual events are defined as follows:

$$\textbf{SR-GUESS}^{\text{key}} : \textbf{GUESS}^{\text{key}} \text{ (of Theorem 1)}, \quad \textbf{SR-COL}^{\text{key}} : \textbf{COL}^{\text{key}} \text{ (of Theorem 1)},$$
$$\textbf{SR-GUESS}_p : \textbf{M-GUESS}_p \text{ (of Theorem 2)}, \quad \textbf{SR-COL}^{\text{aux}} : \textbf{COL}^{\text{aux}} \text{ (of Theorem 1)},$$
$$\textbf{SR-COL}^{\text{st}} : \textbf{M-COL}^{\text{st}} \text{ (of Theorem 2)},$$
$$\textbf{SR-INNER} : \exists (X, Y, \text{inv}, q, 0), (S, S', \text{fwd}, \mathcal{O}, j) \in \tau \text{ such that } X \overset{c}{\in} \texttt{InterSt}((S, S', \text{fwd}, \mathcal{O}, j)).$$

Note that since the adversary has access to all intermediate states, **GUESS**$_q$ (or a variant of this event) is not needed as bad event. The analysis of the bad events in Lemma 5 is similar to that of Lemma 4, with the difference that we use multicollisions in order to upper bound the maximum number of intermediate states that have their leftmost $c - k$ bits of their inner part equal to a certain value. □

Lemma 5. *We have*

$$\begin{aligned}\textbf{Pr}\left(\textbf{SR-BAD}\right) \leq & \binom{\mu}{2}\frac{1}{2^k} + \frac{2\mu(Q_p + Q_{\mathcal{E}} + Q_{\mathcal{D}})}{2^k} + \frac{(12(c-k)+16)Q_p}{2^k} \\ & + \frac{Q_p(6m+8)}{2^{n-m}} + \frac{4Q_p(Q_{\mathcal{E}} + Q_{\mathcal{D}})}{2^n} + \frac{8(2Q_{\mathcal{E}} + 2Q_{\mathcal{D}} + 8Q_q + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}})^2}{2^c} \\ & + \frac{12Q_p(8Q_{\mathcal{E}} + 8Q_{\mathcal{D}} + Q_q + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}})}{2^c}.\end{aligned}$$

The proof can be found in Appendix C. It is very similar to that of Lemma 4, with the main distinction being the additional coverage of the bad event **SR-INNER** and the treatment of direct q queries as construction queries.

5.4.3 Tightness. Additionally to attacks that apply in the "conventional" (i.e., non-state-recovery) nonce-misuse setting, the occurrence of inner collisions between q-evaluations can help the adversary to mount forgeries. Indeed, consider the following adversary:

- It first makes any encryption learning query with an associated data of 1 block (w.l.o.g., already padded): $\mathcal{LE}^{p,q}_{K_1}(N, A)$. In particular, it learns the state S right after absorption of A and before application of permutation p without the domain separator bits;
- It queries $p(S \oplus (A_i \parallel 0^c))$ for $2^{c/2}$ different A_i;[3]
- With high probability, there exist $A_i \neq A_j$ such that $p(S \oplus (A_i \parallel 0^c)) \overset{c}{=} p(S \oplus (A_j \parallel 0^c))$. Let $A_\Delta = \lceil p(S \oplus (A_i \parallel 0^c)) \oplus p(S \oplus (A_j \parallel 0^c)) \rceil_r \oplus (0^{r-1} \parallel 1)$;
- If such a collision occurs, the adversary queries $\mathcal{E}^{p,q}_{K_1}(N, A \parallel A_i \parallel (0^{r-1} \parallel 1))$, and learns the tag T;
- Finally, the adversary outputs a forgery $(N, A \parallel A_j \parallel A_\Delta, T)$.

[3] Note, if $r < c/2$, this attack can be extended by making several sequential absorb calls.

This attack involves finding inner collisions in the sponge construction based on the permutation q with initial state S. The adversary manages to find a full-state collision immediately before the last p-evaluation, thereby obtaining two sequences of blocks that produce the same tag.

6 Conclusion

In this work we derived a security bound on the security of the mode of operation of the Ascon authenticated encryption scheme. Besides the conventional confidentiality and authenticity, we also investigated formally what happens in Ascon if an inner state happens to be recovered by the adversary. Using our tailormade definition, we proved that Ascon even guarantees authenticity in this case, unlike typical duplex-style authenticated encryption modes that only use the key upon initialization. We stress that our results hold in the ideal permutation model. For instance, Baudrin et al. [5] performed a practical attack against the actual Ascon authenticated encryption scheme in the setting of a nonce-misusing adversary, with a complexity of around 2^{40}. This attack exploits weaknesses of the underlying permutation.

Besides an authenticated encryption mode, Ascon also offers a hashing functionality. For completeness, we remark that generic security of this hashing follows from the plain indifferentiability of the sponge [9], guaranteeing $c/2$-bit security. Recently, the developers also introduced Ascon-PRF [23], a pseudorandom function on top of the Ascon permutation. Security of this mode was proven in [33, Section 8.3].

Acknowledgements. We want to thank Christoph Dobraunig for the insightful discussions on the generic security of the Ascon mode and the role of the key blinding. Charlotte Lefevre is supported by the Netherlands Organisation for Scientific Research (NWO) under grant OCENW.KLEIN.435. Bart Mennink is supported by the Netherlands Organisation for Scientific Research (NWO) under grant VI.Vidi.203.099.

References

1. Andreeva, E., Daemen, J., Mennink, B., Van Assche, G.: Security of Keyed Sponge Constructions Using a Modular Proof Approach. In: Leander, G. (ed.) FSE 2015,. Lecture Notes in Computer Science, vol. 9054, pp. 364–384. Springer (2015)
2. Aumasson, J., Henzen, L., Meier, W., Naya-Plasencia, M.: Quark: A Lightweight Hash. In: Mangard, S., Standaert, F. (eds.) CHES 2010. Lecture Notes in Computer Science, vol. 6225, pp. 1–15. Springer (2010)
3. Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT: A Small Present - Towards Reaching the Limit of Lightweight Encryption. In: Fischer, W., Homma, N. (eds.) CHES 2017. Lecture Notes in Computer Science, vol. 10529, pp. 321–345. Springer (2017)

4. Barwell, G., Martin, D.P., Oswald, E., Stam, M.: Authenticated Encryption in the Face of Protocol and Side Channel Leakage. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. Lecture Notes in Computer Science, vol. 10624, pp. 693–723. Springer (2017)
5. Baudrin, J., Canteaut, A., Perrin, L.: Practical cube attack against nonce-misused ascon. ToSC 2022(4), 120–144 (2022)
6. Bellare, M., Namprempre, C.: Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. J. Cryptol. 21(4), 469–491 (2008)
7. Bellare, M., Rogaway, P.: The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. Lecture Notes in Computer Science, vol. 4004, pp. 409–426. Springer (2006)
8. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Sponge functions. Ecrypt Hash Workshop 2007 (May 2007)
9. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the Indifferentiability of the Sponge Construction. In: Smart, N.P. (ed.) EUROCRYPT 2008. Lecture Notes in Computer Science, vol. 4965, pp. 181–197. Springer (2008)
10. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. In: Miri, A., Vaudenay, S. (eds.) SAC 2011. Lecture Notes in Computer Science, vol. 7118, pp. 320–337. Springer (2011)
11. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the Security of the Keyed Sponge Construction. Symmetric Key Encryption Workshop (SKEW 2011) (2011)
12. Bogdanov, A., Knezevic, M., Leander, G., Toz, D., Varici, K., Verbauwhede, I.: Spongent: A Lightweight Hash Function. In: Preneel, B., Takagi, T. (eds.) CHES 2011. Lecture Notes in Computer Science, vol. 6917, pp. 312–325. Springer (2011)
13. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. Lecture Notes in Computer Science, vol. 4727, pp. 450–466. Springer (2007)
14. CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness (May 2014), http://competitions.cr.yp.to/caesar.html
15. Chakraborty, B., Dhar, C., Nandi, M.: Exact security analysis of ASCON. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023. Lecture Notes in Computer Science, vol. 14440, pp. 346–369. Springer (2023)
16. Chakraborty, B., Jha, A., Nandi, M.: On the Security of Sponge-type Authenticated Encryption Modes. ToSC 2020(2), 93–119 (2020)
17. Choi, W., Lee, B., Lee, J.: Indifferentiability of Truncated Random Permutations. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019. Lecture Notes in Computer Science, vol. 11921, pp. 175–195. Springer (2019)
18. Daemen, J., Mennink, B., Van Assche, G.: Full-State Keyed Duplex with Built-In Multi-user Support. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. Lecture Notes in Computer Science, vol. 10625, pp. 606–637. Springer (2017)
19. Daemen, J., Peeters, M., Van Assche, G., Rijmen, V.: Nessie Proposal: NOEKEON. First Open NESSIE Workshop (2000)
20. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1 (2014), submission to CAESAR competition
21. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1.2. Winning Submission to NIST Lightweight Cryptography (2021)
22. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1.2: Lightweight authenticated encryption and hashing. J. Cryptol. 34(3), 33 (2021)

23. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon MAC, PRF, and Short-Input PRF - Lightweight, Fast, and Efficient Pseudorandom Functions. In: Oswald, E. (ed.) CT-RSA 2024. Lecture Notes in Computer Science, vol. 14643, pp. 381–403. Springer (2024)
24. Dobraunig, C., Mennink, B.: Leakage Resilience of the Duplex Construction. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019. Lecture Notes in Computer Science, vol. 11923, pp. 225–255. Springer (2019)
25. Dobraunig, C., Mennink, B.: Security of the Suffix Keyed Sponge. ToSC 2019(4), 223–248 (2019)
26. Guo, C., Pereira, O., Peters, T., Standaert, F.: Towards Low-Energy Leakage-Resistant Authenticated Encryption from the Duplex Sponge Construction. Cryptology ePrint Archive, Report 2019/193 (2019), http://eprint.iacr.org/2019/193 (full version of [27])
27. Guo, C., Pereira, O., Peters, T., Standaert, F.: Towards Low-Energy Leakage-Resistant Authenticated Encryption from the Duplex Sponge Construction. ToSC 2020(1), 6–42 (2020)
28. Guo, J., Peyrin, T., Poschmann, A.: The PHOTON Family of Lightweight Hash Functions. In: Rogaway, P. (ed.) CRYPTO 2011. Lecture Notes in Computer Science, vol. 6841, pp. 222–239. Springer (2011)
29. Hoeffding, W.: Probability inequalities for sums of bounded random variables. In: The collected works of Wassily Hoeffding, pp. 409–426. Springer (1994)
30. 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. Lecture Notes in Computer Science, vol. 8873, pp. 85–104. Springer (2014)
31. Jovanovic, P., Luykx, A., Mennink, B., Sasaki, Y., Yasuda, K.: Beyond Conventional Security in Sponge-Based Authenticated Encryption Modes. J. Cryptol. 32(3), 895–940 (2019)
32. Lefevre, C., Mennink, B.: Generic security of the ascon mode: On the power of key blinding. Cryptology ePrint Archive, Paper 2023/796 (2023), https://eprint.iacr.org/2023/796, https://eprint.iacr.org/2023/796
33. Mennink, B.: Understanding the Duplex and Its Security. ToSC 2023(2), 1–46 (2023)
34. Mennink, B., Reyhanitabar, R., Vizár, D.: Security of Full-State Keyed Sponge and Duplex: Applications to Authenticated Encryption. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. Lecture Notes in Computer Science, vol. 9453, pp. 465–489. Springer (2015)
35. Naito, Y., Yasuda, K.: New Bounds for Keyed Sponges with Extendable Output: Independence Between Capacity and Message Length. In: Peyrin, T. (ed.) FSE 2016,. Lecture Notes in Computer Science, vol. 9783, pp. 3–22. Springer (2016)
36. NIST: Lightweight Cryptography (February 2019), https://csrc.nist.gov/Projects/Lightweight-Cryptography
37. Pereira, O., Standaert, F., Vivek, S.: Leakage-Resilient Authentication and Encryption from Symmetric Cryptographic Primitives. In: Ray, I., Li, N., Kruegel, C. (eds.) ACM SIGSAC 2015. pp. 96–108. ACM (2015)

A Balls-and-Bins Problem

The following lemma upper bounds the expected maximum load of a classical bin-and-balls experiment. It is based on a result of Choi et al. [17] but slightly improved in the fact that it does not have term $\ln(n)$.

Lemma 6 (Choi et al. [17], adapted). *Let $R, n \in \mathbb{N}$. Consider a bin containing balls, where each ball has one of R possible labels and such that there are an equal number of balls for each label. Consider the experiment of sampling n balls in the bin with replacement. For $r \in [\![1, R]\!]$, let X_r be the number of balls drawn with label "r". Then,*

$$\mathsf{E}\left(\max_r X_r\right) \leq \frac{2n}{R} + 3\ln(R) + 4\,.$$

The result also holds when the sampling is performed without replacement.

Proof. Let $p = \frac{1}{R}$ and $r \in [\![1, R]\!]$. Then, X_r follows a binomial law with parameters p and n. Therefore, we can use the Chernoff bound which states that for any $j \geq 2pn$,

$$\mathbf{Pr}\left(X_r \geq j\right) \leq e^{-\frac{j-pn}{3}}\,.$$

Therefore,

$$\begin{aligned}
\mathsf{E}\left(\max_r X_r\right) &= \sum_{j\geq 1} \mathbf{Pr}\left(\max_r X_r \geq j\right) \\
&\leq \sum_{j=1}^{2pn+3\ln(R)} \mathbf{Pr}\left(\max_r X_r \geq j\right) + \sum_{j=2pn+3\ln(R)}^{n} \mathbf{Pr}\left(\bigvee_r X_r \geq j\right) \\
&\leq 2pn + 3\ln(R) + \sum_{r=1}^{R} \sum_{j=2pn+3\ln(R)}^{n} \mathbf{Pr}\left(X_r \geq j\right) \\
&\leq 2pn + 3\ln(R) + R \cdot \sum_{j=2pn+3\ln(R)}^{n} e^{-\frac{j-pn}{3}} \\
&\leq 2pn + 3\ln(R) + R \cdot e^{\frac{pn}{3}} \cdot \frac{e^{-\frac{2pn+3\ln(R)}{3}} - e^{\frac{-n-1}{3}}}{1 - e^{-\frac{1}{3}}} \\
&\leq 2pn + 3\ln(R) + 4R \cdot e^{-\frac{pn}{3}} e^{-\ln(R)} \\
&\leq \frac{2n}{R} + 3\ln(R) + 4\,.
\end{aligned}$$

In addition, when the balls are drawn without replacement, we can apply [29, Theorem 4], which proves that for any function that is both continuous and convex,

$$\mathsf{E}\left(f\left(X_r\right)\right) \leq \mathsf{E}\left(f\left(Y^{(r)}\right)\right)\,,$$

where $Y^{(r)} \sim \text{Binomial}(p, n)$. In particular, this inequality holds when $f(x) = e^{t\cdot x}$ for any $t > 0$. Because the Chernoff bound is obtained by upper bounding $\mathsf{E}\left(e^{t\cdot X_r}\right)$, the proof can be extended to this case. □

B Proof of Lemma 2

Proof (of Lemma 2). For brevity, for an event $\mathbf{evt}$, $\mathbf{Pr}\left(\mathcal{A}^{W_R} \text{ sets } \mathbf{evt}\right)$ will be referred to as $\mathbf{Pr}\left(\mathbf{evt}\right)$. Recall that $\mathbf{BAD} = \mathbf{GUESS}^{\text{key}} \vee \mathbf{GUESS}_p \vee \mathbf{GUESS}_q \vee \mathbf{COL}^{\text{key}} \vee \mathbf{COL}^{\text{aux}} \vee \mathbf{COL}^{\text{st}}$. Let

$$\mathbf{EVT} = \left\{\mathbf{GUESS}^{\text{key}}, \mathbf{GUESS}_p, \mathbf{GUESS}_q, \mathbf{COL}^{\text{aux}}, \mathbf{COL}^{\text{st}}\right\}$$

be the set of all bad events excluding $\mathbf{COL}^{\text{key}}$. We will reason in a primitive-evaluation-wise fashion, and evaluate the probability that $\mathbf{BAD}$ is set at the i^{th} primitive evaluation, conditioned on the fact that it was not set beforehand. There are a total of $Q_p + 2Q_{\mathcal{E}}$ f_p-evaluations and a total of $Q_q + \sigma_{\mathcal{E}}$ f_q-evaluations. Let $v = Q_p + 2Q_{\mathcal{E}} + Q_q + \sigma_{\mathcal{E}}$. For $i \in [\![1, v]\!]$, we denote by $\mathbf{evt}[i]$ the event that $\mathbf{evt}$ is set right after the i^{th} evaluation. Note that $\mathbf{COL}^{\text{key}}$ is the only event that can be set only before the interaction, and for convenience of notation, $\mathbf{BAD}[0]$ refers to $\mathbf{COL}^{\text{key}}$. Moreover, the order at which the evaluations are done or whether the evaluation i is a f_p- or f_q-evaluation does not matter at this point, as this will be addressed later in the proof.

By first splitting $\mathbf{BAD}$ into $\mathbf{BAD}[0] \vee \cdots \vee \mathbf{BAD}[v]$ and subsequently splitting each $\mathbf{BAD}[i]$ into $\vee_{\mathbf{evt} \in \mathbf{EVT}} \mathbf{evt}[i]$, we obtain

$$\mathbf{Pr}\left(\mathbf{BAD}\right) \leq \mathbf{Pr}\left(\mathbf{COL}^{\text{key}}\right) + \sum_{i=1}^{v} \sum_{\mathbf{evt} \in \mathbf{EVT}} \underbrace{\mathbf{Pr}\left(\mathbf{evt}[i] \wedge \neg\mathbf{BAD}[i-1] \wedge \bigwedge_{\mathbf{evt}' \neq \mathbf{evt}} \neg\mathbf{evt}'[i]\right)}_{(8)} . \quad (9)$$

Here, we used the observation that one primitive evaluation (either direct or through the construction) cannot set two different events $\mathbf{evt}$ and $\mathbf{evt}'$ at the same time. Indeed, if this were the case, $\mathbf{GUESS}$ or $\mathbf{COL}$ would have been set in an earlier query. $\mathbf{GUESS}^{\text{key}}$ and $\mathbf{GUESS}_p$ can be set only during f_p-evaluations (either direct or through the construction), $\mathbf{GUESS}_q$ only during f_q-evaluations (either direct or through the construction), and $\mathbf{COL}^{\text{st}}$ only during primitive queries made by construction calls. Let $i \in [\![1, v]\!]$, and denote by $\tau[i]$ the transcript obtained after the first i evaluations.

We will now analyze the probabilities of (9) separately.

Upper Bounding $\mathbf{COL}^{key}$. For $\mathbf{COL}^{\text{key}}$, we have

$$\mathbf{Pr}\left(\mathbf{COL}^{\text{key}}\right) \leq \binom{\mu}{2} \frac{1}{2^k} . \quad (10)$$

Additional Notation for Analyzing (8). We will use the following abbreviation, for any event $\mathbf{evt} \in \mathbf{EVT}$:

$$\mathbf{Cond\_evt}[i] = \neg\mathbf{BAD}[i-1] \wedge \bigwedge_{\mathbf{evt}' \neq \mathbf{evt}} \neg\mathbf{evt}'[i] .$$

Moreover, for $\mathcal{O} \in \{\mathcal{C}, \mathcal{A}\}$, $r \in \{p, q\}$, $\mathbf{1}^i_{\mathcal{O},r}$ denotes the indicator function equal to one if and only if the evaluation number i is an f_r-evaluation and originates from $\mathcal{O}$, where $\mathcal{O} = \mathcal{C}$ denotes the construction and $\mathcal{O} = \mathcal{A}$ the adversary.

We will now upper bound (8) for each $\mathbf{evt} \in \mathbf{EVT}$ separately, for any evaluation number i.

Upper Bounding $\mathbf{COL}^{aux}$. Recall that this event is set when a final state, right before the last p-evaluation collides with a certain $IV \parallel K_j$ on its leftmost $n-m$ bits. These final states are produced either during the first f_p-evaluation (when the associated data is empty and there is at most one block of plaintext), or during the last f_q-evaluation. Because of $\neg\mathbf{BAD}[i-1]$, these states are uniformly random. Therefore

$$\mathbf{Pr}\left(\mathbf{COL}^{\text{aux}}[i] \mid \mathbf{Cond\_COL}^{\text{aux}}[i]\right) \leq \mathbf{1}^{i+1}_{\mathcal{C},p} \frac{\mu}{2^{n-m}}\,. \tag{11}$$

Upper Bounding $\mathbf{COL}^{st}$. First note that, as $\mathbf{COL}^{\text{key}}$ does not occur and the nonces are never repeated for each user, each construction evaluation has a unique starting state $IV \parallel K_j \parallel N$. Furthermore, as long as $\mathbf{GUESS}[i-1]$ and $\mathbf{COL}[i-1]$ do not occur, the primitive evaluation made by the construction during the evaluation number i is fresh. Moreover, as the adversary is not allowed to repeat nonces, it has to commit to the data to absorb before obtaining the ciphertext and tag. Therefore, overwriting the outer parts of the states does not increase the adversarial success probability into setting $\mathbf{COL}^{\text{st}}$. Now, depending on the origin of the evaluation number i, there are several (disjoint) possibilities:

- The evaluation comes from the first f_p-evaluation. Because the keys are not colliding and nonces are always unique per user, all inputs to the first f_p-evaluation are distinct. The evaluation yields to a new random n-bit state (after possibly xoring with some predetermined string). The evaluation sets $\mathbf{COL}^{\text{st}}$ if this state is equal to any earlier inner state. In all cases, this event is set with probability at most $\mathbf{1}^i_{\mathcal{C},p} \dfrac{Q_{\mathcal{E}} + \sigma_{\mathcal{E}}}{2^n}$;
- The evaluation comes from one f_q-evaluation. Depending on the location of this evaluation, the state produced by this iteration yields to a certain intermediate or final state (after possibly xoring with some predetermined string). Similarly to the first case, this event is set with probability at most $\mathbf{1}^i_{\mathcal{C},q} \dfrac{Q_{\mathcal{E}} + \sigma_{\mathcal{E}}}{2^n}$;
- The evaluation comes from the last f_p-evaluation. The state produced plays no role in bad event $\mathbf{COL}^{\text{st}}$.

Therefore,

$$\mathbf{Pr}\left(\mathbf{COL}^{\text{st}}[i] \mid \mathbf{Cond\_COL}^{\text{st}}[i]\right) \leq \left(\mathbf{1}^i_{\mathcal{C},p} + \mathbf{1}^i_{\mathcal{C},q}\right) \frac{Q_{\mathcal{E}} + \sigma_{\mathcal{E}}}{2^n}\,. \tag{12}$$

Upper Bounding $\mathbf{GUESS}^{key}$. This event can be set only with an adversarial f_p-query. In the case where this query is inverse, hitting $IV \parallel K_j$ on its leftmost $n-m$ bits is an unlucky event, which happens with probability at most $\frac{\mu}{2^{n-m}}$. In the case where the query is made in the forward direction, this event is set if the attacker guesses one of the μ uniformly random keys correctly during any of the Q_p queries. Each failed guess eliminates one state from the set of possible candidates. Therefore, one has

$$\mathbf{Pr}\left(\mathbf{GUESS}^{\text{key}}[i] \mid \mathbf{Cond\_GUESS}^{\text{key}}[i]\right) \leq \mathbf{1}^{i}_{\mathcal{A},p} \frac{\mu}{2^k - Q_p} \leq \mathbf{1}^{i}_{\mathcal{A},p} \frac{2\mu}{2^k}\,, \quad (13)$$

where we used $Q_p \leq 2^{k-1}$, and that $k+m \leq n$ so the case of inverse queries is covered.

Upper Bounding $\mathbf{GUESS}_p$ *and* $\mathbf{GUESS}_q$. This event can occur through two scenarios: (i) when the query triggering this event is an inverse permutation query or a forward construction query, or (ii) when the query triggering this event is a forward permutation query. In the former case, the probability is evenly distributed. Therefore, we focus our attention on scenario (ii), where the adversary directly guesses the state. We can observe that the adversary may have knowledge of the leftmost r bits of all states where one block of plaintext is absorbed. In addition, it has access to the rightmost m bits of the final state after the key is xored. The knowledge of these leftmost r bits and rightmost m bits may increase the adversarial success probability, as it can focus on states whose leftmost r bits or rightmost m bits are most-reoccurring, and guessing the remaining c or $n-m$ bits. To capture this situation, we define the following families of random variables:

$$\begin{aligned}
\mathtt{Col_q}[i] &= \max_{z\in\{0,1\}^r} \#\Big\{U \mid \exists (U,V,\text{fwd},\mathcal{E}_q,j) \text{ or } (U,V,\text{fwd},\mathcal{E}_p^{\text{end}},j) \in \tau[i] \text{ such that } \lceil U\rceil_r = z\Big\},\\
\mathtt{Col_p}[i] &= \max_{z\in\{0,1\}^r} \#\Big\{Z \mid \exists (U,V,\text{fwd},\mathcal{E}_p^{\text{beg}},j) \in \tau[i] \text{ such that } Z = V \wedge \lceil Z\rceil_r = z \text{ or }\\
&\qquad\qquad \exists (U,V,\text{fwd},\mathcal{E}_p^{\text{end}},j) \in \tau[i] \text{ such that } Z = U \wedge \lceil Z\rceil_r = z\Big\},\\
\mathtt{Col_t}[i] &= \max_{z\in\{0,1\}^m} \#\Big\{V \mid \exists (U,V,\text{fwd},\mathcal{E}_p^{\text{end}},j) \in \tau[i] \text{ such that } \lfloor V\rfloor_m = z\Big\}. \qquad (14)
\end{aligned}$$

Moreover, we define $\mathtt{Col_x} = \mathtt{Col_x}[v]$. These random variables count the maximum number of jointly colliding ciphertexts/tags without **COL** being set so far. For $\mathtt{Col_q}[i]$ and $\mathtt{Col_p}[i]$, we examine the states before and after each q-evaluation and p-evaluation, respectively. We can now further evaluate (8) for

each $\mathbf{evt} \in \{\mathbf{GUESS}_p, \mathbf{GUESS}_q\}$. We have

$$\mathbf{Pr}\left(\mathbf{GUESS}_q[i] \wedge \mathbf{Cond\_GUESS}_q[i]\right) \leq$$
$$\sum_{\mathtt{c_q}} \mathbf{Pr}\left(\mathbf{GUESS}_q[i] \;\middle|\; \mathbf{Cond\_GUESS}_q[i] \wedge \mathtt{Col_q}[i] = \mathtt{c_q}\right) \times \mathbf{Pr}\left(\mathtt{Col_q}[i] = \mathtt{c_q}\right) , \quad (15)$$
$$\mathbf{Pr}\left(\mathbf{GUESS}_p[i] \wedge \mathbf{Cond\_GUESS}_p[i]\right) \leq$$
$$\sum_{\mathtt{c_p},\mathtt{c_t}} \mathbf{Pr}\left(\mathbf{GUESS}_p[i] \;\middle|\; \mathbf{Cond\_GUESS}_p[i] \wedge \mathtt{Col_p}[i] = \mathtt{c_p} \wedge \mathtt{Col_t}[i] = \mathtt{c_t}\right) \times$$
$$\mathbf{Pr}\left(\mathtt{Col_p}[i] = \mathtt{c_p}\right) \mathbf{Pr}\left(\mathtt{Col_t}[i] = \mathtt{c_t}\right) , \quad (16)$$

where we used that $\mathtt{Col_p}[i]$ and $\mathtt{Col_t}[i]$ are independent. In the following, let $\mathtt{c_q}$, $\mathtt{c_p}$, $\mathtt{c_t}$ be fixed. We can now evaluate the conditioned probabilities of (15) and (16). We start with the conditioned $\mathbf{GUESS}_q[i]$. There are two possibilities to set this event at the iteration number i:

- $\mathbf{GUESS}_q[i]$ is set during an adversarial primitive query. Because of $\neg\mathbf{COL}[i]$, the intermediate states generated during previous construction queries are uniformly random, and independent. By the condition $\mathtt{Col_q}[i] = \mathtt{c_q}$, and $\neg\mathbf{COL}[i]$, there are at most $\mathtt{c_q}$ different states with a given outer part, thus one attempt from the adversary can target at most $\mathtt{c_q}$ states. Moreover, the inner part is uniformly random, and one failed attempt from the adversary only eliminates one state from the list of possible candidates. Therefore, this event is set with probability at most

$$\mathbf{1}^i_{\mathcal{A},q} \frac{\mathtt{c_q}}{2^c - Q_q} + \mathbf{1}^i_{\mathcal{A},q} \frac{Q_\mathcal{E} + \sigma_\mathcal{E}}{2^n} \leq \mathbf{1}^i_{\mathcal{A},q} \frac{2\mathtt{c_q}}{2^c} + \mathbf{1}^i_{\mathcal{A},q} \frac{Q_\mathcal{E} + \sigma_\mathcal{E}}{2^n} ,$$

  where the second term corresponds to an unlucky collision on the output of the adversarial query;
- $\mathbf{GUESS}_q[i]$ is set during a construction query. In that setting, it means that the output of a freshly generated random value appears in the adversary query history. Therefore, this happens with probability at most $\mathbf{1}^i_{\mathcal{C},q} \frac{Q_q}{2^n}$.

Therefore,

$$\mathbf{Pr}\left(\mathbf{GUESS}_q[i] \;\middle|\; \mathbf{Cond\_GUESS}_q[i] \wedge \mathtt{Col_q}[i] = \mathtt{c_q}\right) \leq$$
$$\mathbf{1}^i_{\mathcal{A},q} \frac{2\mathtt{c_q}}{2^c} + \mathbf{1}^i_{\mathcal{A},q} \frac{Q_\mathcal{E} + \sigma_\mathcal{E}}{2^n} + \mathbf{1}^i_{\mathcal{C},q} \frac{Q_q}{2^n} .$$

The conditioned $\mathbf{GUESS}_p[i]$ can be evaluated using the same reasoning. Note that the adversary does not have a direct access to the rightmost m bits of the final state, but this does not change the upper bounding. We obtain

$$\mathbf{Pr}\left(\mathbf{GUESS}_p[i] \;\middle|\; \mathbf{Cond\_GUESS}_p[i] \wedge \mathtt{Col_p}[i] = \mathtt{c_p} \wedge \mathtt{Col_t}[i] = \mathtt{c_t}\right) \leq$$
$$\mathbf{1}^i_{\mathcal{A},p} \frac{2\mathtt{c_p}}{2^c} + \mathbf{1}^i_{\mathcal{A},p} \frac{2\mathtt{c_t}}{2^{n-m}} + \mathbf{1}^i_{\mathcal{A},p} \frac{2Q_\mathcal{E}}{2^n} + \mathbf{1}^i_{\mathcal{C},p} \frac{Q_p}{2^n} .$$

Plugging these into (15) and (16) gives

$$\begin{aligned}
&\mathbf{Pr}\left(\mathbf{GUESS}_q[i] \wedge \mathbf{Cond_GUESS}_q[i]\right) \\
&\leq \sum_{\mathtt{c_q}} \left(\mathbf{1}^i_{\mathcal{A},q}\frac{2\mathtt{c_q}}{2^c} + \mathbf{1}^i_{\mathcal{A},q}\frac{Q_\mathcal{E}+\sigma_\mathcal{E}}{2^n} + \mathbf{1}^i_{\mathcal{C},q}\frac{Q_q}{2^n}\right)\mathbf{Pr}\left(\mathtt{Col_q}[i]=\mathtt{c_q}\right) \\
&= \mathbf{1}^i_{\mathcal{A},q}\frac{2\mathsf{E}\left(\mathtt{Col_q}\right)}{2^c} + \mathbf{1}^i_{\mathcal{A},q}\frac{Q_\mathcal{E}+\sigma_\mathcal{E}}{2^n} + \mathbf{1}^i_{\mathcal{C},q}\frac{Q_q}{2^n}\,, \qquad (17)
\end{aligned}$$

and

$$\begin{aligned}
&\mathbf{Pr}\left(\mathbf{GUESS}_p[i] \wedge \mathbf{Cond_GUESS}_p[i]\right) \leq \\
&\qquad \mathbf{1}^i_{\mathcal{A},p}\frac{2\mathsf{E}\left(\mathtt{Col_p}\right)}{2^c} + \mathbf{1}^i_{\mathcal{A},p}\frac{2\mathsf{E}\left(\mathtt{Col_t}\right)}{2^{n-m}} + \mathbf{1}^i_{\mathcal{A},p}\frac{2Q_\mathcal{E}}{2^n} + \mathbf{1}^i_{\mathcal{C},p}\frac{Q_p}{2^n}\,. \qquad (18)
\end{aligned}$$

Conclusion. By plugging (10), (11), (12), (13), (17), and (18) into (9), we obtain

$$\begin{aligned}
\mathbf{Pr}\left(\mathbf{BAD}\right) \leq \binom{\mu}{2}\frac{1}{2^k} + \frac{2\mu Q_p}{2^k} + \frac{2\mu Q_\mathcal{E}}{2^{n-m}} + \frac{\left(2Q_\mathcal{E}+\sigma_\mathcal{E}\right)^2}{2^n} + \frac{Q_q\left(3Q_\mathcal{E}+2\sigma_\mathcal{E}\right)}{2^n} + \\
\frac{2Q_pQ_\mathcal{E}}{2^n} + \frac{2Q_q\mathsf{E}\left(\mathtt{Col_q}\right)}{2^c} + \frac{2Q_p\mathsf{E}\left(\mathtt{Col_p}\right)}{2^c} + \frac{2Q_p\mathsf{E}\left(\mathtt{Col_t}\right)}{2^{n-m}}\,, \qquad (19)
\end{aligned}$$

where we used that

$$\begin{aligned}
&\#\left\{i \in [\![1,v]\!] \mid \mathbf{1}^i_{\mathcal{C},p}=1\right\} = 2Q_\mathcal{E}\,, \qquad &&\#\left\{i \in [\![1,v]\!] \mid \mathbf{1}^i_{\mathcal{A},q}=1\right\} = Q_q\,, \\
&\#\left\{i \in [\![1,v]\!] \mid \mathbf{1}^i_{\mathcal{C},q}=1\right\} = \sigma_\mathcal{E}\,, &&\#\left\{i \in [\![1,v]\!] \mid \mathbf{1}^i_{\mathcal{A},p}=1\right\} = Q_p\,.
\end{aligned}$$

It remains to bound the expectations of $\mathtt{Col_q}$, $\mathtt{Col_p}$, and $\mathtt{Col_t}$. These random variables count the maximum number of colliding states that come from distinct primitive evaluations, where we note that $\mathtt{Col_p}$ can be upper bounded as the sum of two sets with independent randomness. Therefore, the situation reduces to the bin-and-balls setting from Lemma 6, and we obtain

$$\begin{aligned}
&\mathsf{E}\left(\mathtt{Col_q}\right) \leq \frac{2(\sigma_\mathcal{E}+Q_\mathcal{E})}{2^r} + 3r + 4\,, \qquad &&\mathsf{E}\left(\mathtt{Col_p}\right) \leq \frac{4Q_\mathcal{E}}{2^r} + 6r + 8\,, \\
&\mathsf{E}\left(\mathtt{Col_t}\right) \leq \frac{2Q_\mathcal{E}}{2^m} + 3m + 4\,,
\end{aligned}$$

where we implicitly used that there are at most $\sigma_\mathcal{E} + Q_\mathcal{E}$ states V_l. Finally, we obtain from (19):

$$\begin{aligned}
\mathbf{Pr}\left(\mathbf{BAD}\right) \leq \binom{\mu}{2}\frac{1}{2^k} + \frac{2\mu Q_p}{2^k} + \frac{2\mu Q_\mathcal{E}}{2^{n-m}} + \frac{\left(2Q_\mathcal{E}+\sigma_\mathcal{E}\right)^2}{2^n} + \frac{12Q_q\left(Q_\mathcal{E}+\sigma_\mathcal{E}\right)}{2^n} + \\
\frac{16Q_pQ_\mathcal{E}}{2^n} + \frac{(6r+8)Q_q}{2^c} + \frac{(12r+16)Q_p}{2^c} + \frac{(6m+8)Q_p}{2^{n-m}}\,.
\end{aligned}$$

This completes the proof. □

C Proof of Lemma 5

In this section, we provide a proof of the upper bound that **SR-BAD** occurs in the case of security under state recovery. The proof is similar to that of Lemma 4, with the main distinction being the additional coverage of the bad event **SR-INNER** and the treatment of direct q queries as construction queries. Let

$$\textbf{SR-EVT} = \left\{\textbf{SR-GUESS}^{\text{key}}, \textbf{SR-GUESS}_p, \textbf{SR-COL}^{\text{aux}}, \textbf{SR-COL}^{\text{st}}, \textbf{SR-INNER}\right\},$$

be the set of all bad events in both settings, excluding $\textbf{SR-COL}^{\text{key}}$. Let $v = 2Q_{\mathcal{E}} + 2Q_{\mathcal{D}} + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}} + Q_p + Q_q$ be the total number of evaluations. We have

$$\textbf{Pr}\left(\textbf{SR-BAD}\right) \leq \textbf{Pr}\left(\textbf{SR-COL}^{\text{key}}\right) + \sum_{i=1}^{v} \sum_{\textbf{evt} \in \textbf{SR-EVT}} \underbrace{\textbf{Pr}\left(\textbf{evt}[i] \wedge \neg\textbf{SR-BAD}[i-1] \wedge \bigwedge_{\textbf{evt}' \neq \textbf{evt}} \neg\textbf{evt}'[i]\right)}_{(20)}. \quad (21)$$

In the following, let $i \in [\![1, v]\!]$. We upper bound the probability of (20) for any $\textbf{evt} \in \textbf{SR-EVT}$.

Identical Events. The bounding of $\textbf{SR-COL}^{\text{key}} = \textbf{COL}^{\text{key}}$, $\textbf{SR-GUESS}^{\text{key}} = \textbf{GUESS}^{\text{key}}$, and $\textbf{SR-COL}^{\text{aux}} = \textbf{COL}^{\text{aux}}$ remains unchanged compared to Lemma 4:

$$\textbf{Pr}\left(\textbf{SR-COL}^{\text{key}}\right) \leq \binom{\mu}{2}\frac{1}{2^k}, \quad (22)$$

$$\textbf{Pr}\left(\textbf{SR-GUESS}^{\text{key}}[i] \mid \textbf{Cond\_SR-GUESS}^{\text{key}}[i]\right) \leq \mathbf{1}^i_{\mathcal{A},p}\frac{2\mu}{2^k}, \quad (23)$$

$$\textbf{Pr}\left(\textbf{SR-COL}^{\text{aux}}[i] \mid \textbf{Cond\_SR-COL}^{\text{aux}}[i]\right) \leq \mathbf{1}^i_{\mathcal{C},p}\frac{\mu}{2^k}. \quad (24)$$

Updated Events. We start with **SR-INNER**. There are two possibilities to set this event: either a candidate intermediate state that was generated during a construction query collides on its inner part with a former inverse query output, or an inverse query collides on its inner part with a candidate intermediate state. As, in current case, there are at most $2(Q_{\mathcal{E}} + Q_{\mathcal{D}} + Q_q + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}})$ candidate intermediate or final states, the event is set with probability at most

$$\textbf{Pr}\left(\textbf{SR-INNER}[i] \mid \textbf{Cond\_SR-INNER}[i]\right) \leq (\mathbf{1}^i_{\mathcal{C},q} + \mathbf{1}^i_{\mathcal{C},p})\frac{4Q_q}{2^c} + \mathbf{1}^i_{\mathcal{A},q}\frac{4(Q_{\mathcal{E}} + Q_{\mathcal{D}} + Q_q + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}})}{2^c}. \quad (25)$$

Next, we consider event $\textbf{SR-COL}^{\text{st}}$. This event can be upper bounded in a similar way as in the nonce-misuse setting of Theorem 2, with a crucial difference

that q-queries from the adversary are now counted as associated to candidate intermediate or final states. Therefore,

$$\mathbf{Pr}\left(\mathbf{SR\text{-}COL}^{\mathrm{st}}[i] \mid \mathbf{Cond\_SR\text{-}COL}^{\mathrm{st}}[i]\right) \leq (\mathbf{1}^i_{\mathcal{C},p} + \mathbf{1}^i_{\mathcal{C},q}) \frac{8(Q_{\mathcal{E}} + Q_{\mathcal{D}} + Q_q + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}})}{2^c}\,. \quad (26)$$

Finally, we consider $\mathbf{SR\text{-}GUESS}_p$. The adversary has two options to trigger this event. Firstly, it can guess the final state from which the tag is extracted. Secondly, it can guess either the output of the first p-evaluation S_1 or the input to the last p-evaluation S_2 that occurred during a construction query. Handling the first case follows a similar approach as in Lemma 4. However, the second case is significantly different from before. In this scenario, the adversary has additional access to the leftmost $c-k$ bits of the inner part of the states S_1 or S_2. Assuming no bad event occurred prior to this, the states S_1 originate from distinct p-evaluations, while the states S_2 come from distinct p- or q-evaluations. These states are randomly sampled in a permutation-consistent way, allowing us to employ multicollisions. We define the following sets:

$$\mathtt{Col_s} = \max_{z\in\{0,1\}^{c-k}} \#\Big\{ Z \mid \exists (U,V,\mathrm{fwd},\mathcal{C}^{\mathrm{beg}}_p, j) \in \tau \text{ such that } Z = V \text{ and} \lceil \lfloor Z \rfloor_c \rceil_{c-k} = z \\ \text{or } \exists (U,V,\mathrm{fwd},\mathcal{O}, j) \in \tau[i] \text{ such that} \\ Z \in \mathtt{LastSt}((U,V,\mathrm{fwd},\mathcal{O},j)) \in \tau \text{ and } \lceil \lfloor Z \rfloor_c \rceil_{c-k} = z \Big\}$$

$$\mathtt{Col_t}[i] = \max_{z\in\{0,1\}^{m}} \#\Big\{ V \mid \exists (U,V,\mathrm{fwd},\mathcal{C}^{\mathrm{end}}_p, j) \in \tau[i] \text{ such that } \wedge \lfloor V \rfloor_m = z \Big\}\,.$$

The expectation of these random variables can be upper bounded using Lemma 6 as

$$\mathsf{E}\left(\mathtt{Col_s}\right) \leq \frac{6(Q_{\mathcal{E}} + Q_{\mathcal{D}}) + 4(\sigma_{\mathcal{E}} + \sigma_{\mathcal{D}} + Q_q)}{2^{c-k}} + 6(c-k) + 8\,,$$
$$\mathsf{E}\left(\mathtt{Col_t}\right) \leq \frac{2(Q_{\mathcal{E}} + Q_{\mathcal{D}})}{2^m} + 3m + 4\,.$$

Thus, one attempt of the adversary to guess one of the states S_1 and S_2, conditioned on $\mathtt{Col_s} = \mathtt{c_s}$ cannot target more than $\mathtt{c_s}$ states at the same time. Moreover, since no bad event happened before, each of these states is randomly generated, in a permutation-consistent way. Therefore,

$$\begin{aligned}&\mathbf{Pr}\left(\mathbf{SR\text{-}GUESS}_p \wedge \mathbf{Cond\_SR\text{-}GUESS}_p[i]\right) \\ &\leq \mathbf{1}^i_{\mathcal{A},p} \frac{4(Q_{\mathcal{E}} + Q_{\mathcal{D}})}{2^n} + \mathbf{1}^i_{\mathcal{A},p} \frac{6m+8}{2^{n-m}} + \mathbf{1}^i_{\mathcal{A},p} \frac{12(c-k)+16}{2^k} + \mathbf{1}^i_{\mathcal{C},p} \frac{2Q_p}{2^c} \\ &+ \mathbf{1}^i_{\mathcal{A},p} \frac{12(Q_{\mathcal{E}} + Q_{\mathcal{D}} + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}} + Q_q)}{2^c}\,. \end{aligned} \quad (27)$$

Conclusion. By plugging (22), (23), (24), (25), (26), and (27) into (21), this completes the proof.