IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2023, No. 1, pp. 244–287. DOI:10.46586/tosc.v2023.i1.244-287

# Chosen-Key Secure Even-Mansour Cipher from a Single Permutation

Shanjie Xu[1,2], Qi Da[1,2] and Chun Guo[1,2,3](✉)

[1] Key Laboratory of Cryptologic Technology and Information Security of Ministry of Education, Shandong University, Qingdao, Shandong, China
[2] School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, China
shanjie1997@mail.sdu.edu.cn, daqi@mail.sdu.edu.cn, chun.guo@sdu.edu.cn
[3] Shandong Research Institute of Industrial Technology, Jinan, Shandong, China

**Abstract.** At EUROCRYPT 2015, Cogliati and Seurin proved that the 4-round Iterated Even-Mansour (IEM) cipher with *Independent random Permutations and no key schedule* $\mathrm{EMIP}_4(k,u) = k \oplus \mathbf{p}_4\big(k \oplus \mathbf{p}_3\big(k \oplus \mathbf{p}_2(k \oplus \mathbf{p}_1(k \oplus u))\big)\big)$ is *sequentially indifferentiable* from an ideal cipher, which implies chosen-key security in the sense of *correlation intractability*. In practice, however, blockciphers such as the AES typically employ the same permutation at each round. To bridge the gap, we prove that the 4-round IEM cipher $\mathrm{EMSP}[\varphi]_4^{\mathbf{P}}(k,u) = k_4 \oplus \mathbf{p}\big(k_3 \oplus \mathbf{p}\big(k_2 \oplus \mathbf{p}(k_1 \oplus \mathbf{p}(k_0 \oplus u))\big)\big)$, whose round keys $k_i = \varphi^i(k)$ are derived using an affine permutation $\varphi : \{0,1\}^n \to \{0,1\}^n$ with certain properties, is *sequentially indifferentiable* from an ideal cipher. The function $\varphi$ can be a linear orthomorphism, or $\varphi(k) := k \ggg_a$ for some fixed integer $a$ using cyclic shift. To our knowledge, this is the first indifferentiability-type result for blockciphers using identical round functions.

**Keywords:** blockcipher · sequential indifferentiability · Even-Mansour cipher

## 1 Introduction

The Iterated Even-Mansour (IEM) scheme (a.k.a. key-alternating cipher), initiated in [EM97] and extended in [GPPR11, BKL+12, DKS12, ABD+13], provides a natural solution to the fundamental cryptographic problem of constructing blockciphers from keyless permutations. Given $t$ permutations $\mathbf{p}_1, ..., \mathbf{p}_t : \{0,1\}^n \to \{0,1\}^n$ and a *key schedule* $\overrightarrow{\varphi} = (\varphi_0, ..., \varphi_t)$, $\varphi_i : \{0,1\}^\kappa \to \{0,1\}^n$, the scheme enciphers $(k,u) \in \{0,1\}^\kappa \times \{0,1\}^n$ as

$$\mathrm{EM}[\overrightarrow{\varphi}]_t(k,u) := \varphi_t(k) \oplus \mathbf{p}_t\big(...\varphi_2(k) \oplus \mathbf{p}_2\big(\varphi_1(k) \oplus \mathbf{p}_1(\varphi_0(k) \oplus u)\big)...\big).$$

It abstracts the paradigm *substitution-permutation network* of a number of standards [Pub01, ISO12, ISO21]. Modeling $\mathbf{p}_1, ..., \mathbf{p}_t$ as public random permutations, this scheme provably achieves various security notions, including indistinguishability [EM97, BKL+12, LPS12, CS14, CLL+18, ML15, HT16, TZ21], related-key security [FP15, CS15], known-key security [ABM14, CS16], chosen-key security [CS15] and indifferentiability [ABD+13, LS13, DSST17]. Despite theoretical uninstantiatability [CGH04, Bla06], such arguments dismiss generic attacks and are viewed as evidences of the soundness of the design approaches.

**Indifferentiability.** The classical security definition for a blockcipher is *indistinguishability from a (secret) random permutation*. Since 2005, a series of papers [CHK+16, ABD+13] have proposed *indifferentiability from ideal ciphers* as a much stronger criteria for ideal function-based blockciphers. Briefly speaking, for the IEM cipher $\mathrm{EM}^{\mathcal{P}}$ built upon random permutations $\mathcal{P}$, if there exists an efficient simulator $\mathcal{S}^E$ that queries an ideal cipher $E$ to

Received: 2022-11-23 Accepted: 2023-01-23 Published: 2023-03-10

mimic its (non-existent) underlying permutations such that $(E, \mathcal{S}^E)$ is indistinguishable from $(\mathrm{EM}^{\mathcal{P}}, \mathcal{P})$, then $\mathrm{EM}^{\mathcal{P}}$ is indifferentiable from $E$ [MRH04]. This property implies that the cipher $\mathrm{EM}^{\mathcal{P}}$ inherits all ideal cipher-properties defined by single-stage games, including security against various forms of related-key and chosen-key attacks.

As results, Andreeva et al. [ABD$^+$13] proposed the IEM variant $\mathrm{EMKD}_t(k,u) = \mathbf{h}(k) \oplus \mathbf{p}_t(...\mathbf{h}(k) \oplus \mathbf{p}_2(\mathbf{h}(k) \oplus \mathbf{p}_1(\mathbf{h}(k) \oplus u))...)$ using a Random Oracle (RO) $\mathbf{h} : \{0,1\}^\kappa \to \{0,1\}^n$ to derive the round key $\mathbf{h}(k)$, and proved indifferentiability at 5 rounds. Concurrently, Lampe and Seurin [LS13] proposed to consider the *"single-key" Even-Mansour* variant $\mathrm{EMIP}_t(k,u) = k \oplus \mathbf{p}_t(...k \oplus \mathbf{p}_2(k \oplus \mathbf{p}_1(k \oplus u))...)$ without any non-trivial key schedule, and proved indifferentiability at 12 rounds. Both results have been tightened in subsequent works [DSST17, GL16a], showing that 3-round EMKD and 5-round EMIP suffice.

**Seq-Indifferentiability and Correlation Intractability.** Known-key and chosen-key attacks typically tried to find tuples of inputs/outputs of the blockcipher that satisfy certain *evasive relations* [CGH04]. For example, the first known-key distinguisher against 7-round Feistel cipher $\Psi_7$ [KR07] attacks by exhibiting a pair of inputs/outputs $\Psi_7(u) = v$ and $\Psi_7(u') = v'$ such that the xor of the right halves of $u, v, u'$ and $v'$ is 0, while the first chosen-key distinguisher against AES-256 [BKN09] attacks by exhibiting $q$-multicollisions.

To formalize known-key security, Knudsen and Rijmen [KR07] discussed the use of *Correlation Intractability (CI)*, meaning that no adversary can find blockcipher inputs/outputs that satisfy evasive relations. Though, this idea was limited by the uninstantiatability of CI (in the standard model) [CGH04], and it gained attention only when positive results were given in the ideal model [MPS12, CS15, CS16]. In detail, to establish CI, Mandal et al. [MPS12] introduced *sequential-indifferentiability* (*seq-indifferentiability*), which is a weaker variant of indifferentiability concentrating on distinguishers that follow a strict restriction on the order of queries. Mandal et al. [MPS12] showed that seq-indifferentiability implies CI, and further established CI for Feistel-based blockciphers. For the aforementioned Even-Mansour variants, CI and seq-indifferentiability have been established for 3-round EMKD [GL16b] and 4-round EMIP [CS15], both of which are tight.

## 1.1 Our Contributions

We stress that all the aforementioned (seq-)indifferentiability results on IEM [ABD$^+$13, LS13, CS15, GL16b, DSST17, GL16a] crucially rely on *using $t$ independent random permutations in the $t$ rounds*. A natural next step is to investigate whether indifferentiability-type security is achievable using a single permutation and (hopefully) minimal key schedule. This is particularly relevant to "real-world" SPN ciphers that typically employ *identical round permutations*, and this has motivated similar attempts regarding indistinguishability [DKS12, CLL$^+$18, Dut20, WYCD20]. Though, due to the significantly added complexity, it remains open to prove indifferentiability with identical permutations.

Arguably, the minimal IEM variant is $\mathrm{EMSP}_t(k,u) = k \oplus \mathbf{p}(...k \oplus \mathbf{p}(k \oplus \mathbf{p}(k \oplus u))...)$ using a single permutation and trivial key schedule. Unfortunately, this is insecure even w.r.t. seq-indifferentiability [XDG23]: by acting as the involved evaluations in all the $t$ rounds, a single permutation-evaluation $\mathbf{p}(x) = y$ already yields a full $t$-round $\mathrm{EMSP}_t$ enciphering $y \to \underbrace{(x,y) \to ... \to (x,y)}_{t \text{ times}} \to x$ with $k = x \oplus y$. Xu et al. [XDG23] thus only proved seq-indifferentiability for a 4-round IEM variant using 2 permutations, which still falls short of addressing the single permutation problem.

Given Xu et al.'s attack, one has to enhance EMSP with some non-trivial key schedules to break the slide property. We seek for using *affine* key schedules due to their popularity in practice. In this respect, we observe that Andreeva et al.'s attack [ABD$^+$13] (see Appendix A) essentially breaks seq-indifferentiability of *any* 3-round IEM with affine key schedules. We thus focus on *4-round IEMs using affine key schedules*.

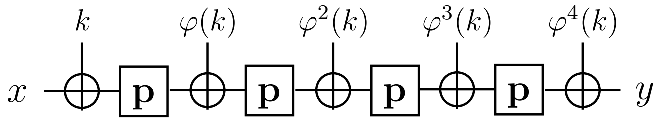

**Figure 1:** The minimal construction $\text{EMSP}[\varphi]_4^{\mathbf{p}}$ using a single random permutation $\mathbf{p} : \{0,1\}^n \to \{0,1\}^n$ and an affine key schedule permutation $\varphi : \{0,1\}^n \to \{0,1\}^n$. One can set $\varphi$ to be a linear orthomorphism, or $\varphi(k) := k \ggg_a$ for a fixed integer $a$.

**Table 1:** Comparison of our result with existing seq-indifferentiable IEM results. The column **Key sch.** indicates the key schedule functions in the schemes. The column **Complex.** indicates the simulator complexities. For $\text{EMSP}[\varphi]_4^{\mathbf{p}}$, $C(\varphi) = 1$ for linear orthomorphism $\varphi$, and $C(\varphi) = 2^a$ for $\varphi(k) = k \ggg_a$, $1 \le a \le n/2$.

| **Scheme** | **♯Rounds** | **♯Primitives** | **Key sch.** | **Complex.** | **Bounds** | **Ref.** |
|---|---|---|---|---|---|---|
| $\text{EMIP}_4^{\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\mathbf{p}_4}$ | 4 | 4 | no | $q^2$ | $q^4/2^n$ | [CS15] |
| $\text{EMKD}_3^{\mathbf{h},\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3}$ | 3 | 4 | RO **h** | $q^2$ | $q^4/2^n$ | [GL16b] |
| $\text{EM2P}_4^{\mathbf{p}_1,\mathbf{p}_2}$ | 4 | **2** | no | $q^2$ | $q^4/2^n$ | [XDG23] |
| $\text{EMSP}[\varphi]_4^{\mathbf{p}}$ | 4 | **1** | iterative | $q^2$ | $C(\varphi)q^7/2^n$ <br> $+q^{10}/2^n$ | This paper |

Our final construction $\text{EMSP}[\varphi]_4^{\mathbf{p}}$ is a 4-round IEM cipher built upon a single permutation and employs an iterative-type key schedule to dissolve the slide property. In detail, let $\varphi : \{0,1\}^n \to \{0,1\}^n$ be a (non-idealized) affine permutation, and denote by $\varphi^r$ the $r$-fold self-composition of $\varphi$. Then, $\text{EMSP}[\varphi]_4^{\mathbf{p}}$, illustrated in Fig. 1, is defined as

$$\text{EMSP}[\varphi]_4^{\mathbf{p}}(k,u) := \varphi^4(k) \oplus \mathbf{p}\big(\varphi^3(k) \oplus \mathbf{p}\big(\varphi^2(k) \oplus \mathbf{p}(\varphi(k) \oplus \mathbf{p}(k \oplus u))\big)\big). \tag{1}$$

We require that for *any* $z \in \{0,1\}^n$, the number of $x \in \{0,1\}^n$ such that $x \oplus \varphi(x) = z$ is at most $C(\varphi)$, and $C(\varphi)/2^n$ is sufficiently small. For instance, if $\varphi$ is a linear orthomorphism, then $x \mapsto x \oplus \varphi(x)$ is also a permutation, and $C(\varphi) = 1$. If $\varphi(k) := k \ggg_a$, i.e., right rotating $k$ by $a$ bits, $1 \le a \le n/2$, then $C(\varphi) = 2^a$. By this, a single input/output $\mathbf{p}(x) = y$ cannot yield a full $\text{EMSP}[\varphi]_4^{\mathbf{p}}$ enciphering any more. We refer to *Technical challenges* below for more details. Formally, we prove that the 4-round $\text{EMSP}[\varphi]_4^{\mathbf{p}}$ is seq-indifferentiable:

**Theorem 1.** *Assume that* $\mathbf{p}$ *is a random permutation,* $3q^2 \le 2^n/2$*. Then, the 4-round Even-Mansour scheme* $EMSP[\varphi]_4^{\mathbf{p}}$ *in Eq.* (1) *is strongly and statistically* $(q, \sigma, t, \varepsilon)$*-seq-indifferentiable from an ideal cipher* $E$*, where* $\sigma = q^2$*,* $t = O(q^2)$*, and* $\varepsilon \le \big(1524C(\varphi)q^7 + 2725q^{10}\big)/2^n$*.*

$\text{EMSP}[\varphi]_4^{\mathbf{p}}$ thus offers the minimal Even-Mansour scheme for chosen-key security in the sense of seq-indifferentiability. Please see Table 1 for comparison.

Remark. We are *not* advocating $\text{EMSP}[\varphi]_4^{\mathbf{p}}$ for practice: $\text{EMSP}[\varphi]_4^{\mathbf{p}}$ is *not* fully indifferentiable, and its security bound is weak. Thus, when instantiating EMSP for CI only, one has to use moderately large permutations (and probably more than 4 rounds). Though, our focus here is on theoretical side, showing feasibility results on achieving (weaker) indifferentiability with identical round functions. Another interesting point is that, our key schedule is "tight" in the sense that it is "just" sufficient for CI! We believe this sheds lights on which properties on the key schedule are needed for blockcipher security.

**Technical Challenges.** We first recall an existing simulator for $\text{EMIP}_4$, which is our basis. We then discuss $\text{EMSP}[\varphi]_4$ and elaborate on challenges, intuitions and our solutions.

*Simulator for EMIP$_4$.* To establish indifferentiability-type security, the first step is to construct a simulator that resists obvious attacks. Then, it remains to argue: (i) The simulator is efficient, i.e., its complexity can be bounded; (ii) The simulator gives rise to an ideal world $(E, \mathcal{S}^E)$ that is indistinguishable from the real world $(\mathrm{EM}^{\mathcal{P}}, \mathcal{P})$.

Virtually all blockcipher simulators follow the *(computation) chain detection and completion* approach initiated by Coron et al. [CHK+16]. For example, consider the EMIP$_4$ cipher (using *independent* permutations). A distinguisher $D$ may arbitrarily pick $k, u \in \{0,1\}^n$ and evaluate $x_1 \leftarrow k \oplus u$, $P_1(x_1) \rightarrow y_1$, $x_2 \leftarrow k \oplus y_1$, $P_2(x_2) \rightarrow y_2$, $x_3 \leftarrow k \oplus y_2$, $P_3(x_3) \rightarrow y_3$, $x_4 \leftarrow k \oplus y_4$, $P_4(x_4) \rightarrow y_4$, $x_5 \leftarrow k \oplus y_4$. This creates a sequence of four *(query) records* $\big((1, x_1, y_1), (2, x_2, y_2), (3, x_3, y_3), (4, x_4, y_4)\big)$ that will be called a *computation chain* (the numbers 1,..., 4 indicates the index of the permutation). When $D$ is in the real world, it necessarily holds $\mathrm{EMIP}_4(k, u) = x_5$.

To be consistent with this in the ideal world, $\mathcal{S}$ should pre-emptively define some simulated (query) records to "complete" a similar chain. To this end, Cogliati and Seurin (CS)'s simulator $\mathcal{S}$ for EMIP$_4$ [CS15] takes queries to the middle (2nd and 3rd) rounds as "signals" for chain detection and the outer (1st and 4th) rounds for adaptations. Concretely, facing the above attack, $\mathcal{S}$ pinpoints the key $k = y_2 \oplus x_3$ and recognize the "partial chain" $\big((1, x_1, y_1), (2, x_2, y_2), (3, x_3, y_3)\big)$ upon the third permutation query $P_2(x_3) \rightarrow y_3$. $\mathcal{S}$ then queries the ideal cipher $E(k, k \oplus x_1) \rightarrow x_5$ and *adapts* the simulated $P_1$ by enforcing $P_1(k \oplus y_3) := k \oplus x_5$. As such, a simulated computation chain $\big((1, x_1, y_1), (2, x_2, y_2), (3, x_3, y_3), (4, k \oplus y_3, k \oplus x_5)\big)$ with $E(k, k \oplus x_1) = x_5$ is completed. Worth noting, adaptations only create records on $P_1$ and $P_4$ and won't trigger new detection. This idea of assigning a unique role to every round/simulated primitive was initiated in [CHK+16], and it significantly simplifies arguments.

Of course, $D$ may pick $k', y'_4 \in \{0,1\}^n$ and evaluate "conversely". In this case, CS simulator detects the "partial chain" $\big((2, x'_2, y'_2), (3, x'_3, y'_3), (4, x'_4, y'_4)\big)$ after $D$'s third query $P_2^{-1}(y'_2) \rightarrow x'_2$, queries $E^{-1}(k', k' \oplus y'_4) \rightarrow x'_0$ and pre-enforces $P_1(k' \oplus x'_0) := k' \oplus x'_5$. In the seq-indifferentiability setting, these have covered all adversarial possibilities. In particular, the distinguisher $D$ cannot pick $k', y'_1$ and evaluate $P_1^{-1}(y'_1) \rightarrow x'_1$, $u' \leftarrow k' \oplus x'_1$, $E(k', u') \rightarrow v'$, and $P_4^{-1}(k' \oplus v') \rightarrow x'_4$, since this violates the query restriction. This greatly simplifies simulation compared with the "full" indifferentiability setting.

*Challenges in EMSP$[\varphi]_4$.* The problem in EMSP$[\varphi]_4$ is that there is no independence at all, and every permutation query $P(x) \rightarrow y$ is potentially in 2nd or 3rd round. If we simply detect and complete all pairs of records, then the simulation runs into an endless recursion: for every detected "2-chain" $\big((x, y), (x', y')\big)$ (since there is only 1 permutation, we omit the index), the simulator has to create at least 1 record $(x'', y'')$ to adapt; but then, the new record $(x'', y'')$ gives rise to new "2-chains" $\big((x, y), (x'', y'')\big)$, $\big((x', y'), (x'', y'')\big)$, ..., and simulator will complete these "2-chains" to create many new "adapted" records, which in turn give rise to new "2-chains" as well. Likely, this is the major obstacle to indifferentiability-type proofs for blockciphers using identical primitives.

*Intuitions for EMSP$[\varphi]_4$.* To rescue, our intuition is that *(query) records created by internal simulator actions are "private" and "unknown" to the distinguisher $D$*, and it is *not* necessary to immediately complete chains for such internal records.

For example, consider $D$ querying $P(x_1)$. $\mathcal{S}$ samples $y_1 \xleftarrow{\$} \{0,1\}^n$ and creates the record $(x_1, y_1)$. This constitutes a "2-chain" $\big((x_1, y_1), (x_1, y_1)\big)$ that has:

- The "next" value is $x'_4 = y_1 \oplus \varphi(y_1 \oplus x_1) = (y_1 \oplus \varphi(y_1)) \oplus \varphi(x_1)$, and the probability to have $x'_4 = x^\circ$ for any "target value" $x^\circ$ is at most $C(\varphi)/2^n$;
- The "previous" value $y'_0 = x_1 \oplus \varphi^{-1}(y_1 \oplus x_1)$ is uniformly distributed in $\{0,1\}^n$.

By these, $\mathcal{S}$ could (roughly speaking): (1) sample $y'_4 \xleftarrow{\$} \{0,1\}^n$ and create a record $(x'_4, y'_4)$; (2) query $E^{-1}\big(\varphi^{-2}(y_1 \oplus x_1), \varphi^2(y_1 \oplus x_1) \oplus y'_4\big) \rightarrow u'$ and create $(x'_0, y'_0)$, $x'_0 :=$

$\varphi^{-2}(y_1 \oplus x_1) \oplus u'$ to adapt. The records $(x'_0, y'_0)$ and $(x'_4, y'_4)$ are "internal created", and are "private". Each such private record has one "endpoint" known to $D$ while the other "unknown" to $D$. Concretely, $y'_1$ and $x'_4$ can be derived from $x_1$ and $y_1$ that are "known" to $D$, but $x'_1$ and $y'_4$ are "unknown" to $D$. Furthermore,

- $y'_4$ is sampled "internally", and appears uniformly distributed in $\{0,1\}^n$ in the view of $D$, till $D$ obtains $y'_4$ by explicitly querying $P(x_4)$;
- $x'_0$ is derived from $u'$ the response of an ideal cipher query, and also appears (somewhat) uniform in the view of $D$, till $D$ explicitly queries $P^{-1}(y_0)$.

Thus, (intuitively) $D$ is unlikely to "guess" the values of $y'_4$ and $x'_0$—or even values that bear "interesting" relations with $y'_4$ and $x'_0$—and use this information in attack. Worth noting, (as discussed) $D$ cannot derive $y'_4$ by querying the ideal cipher $E$ itself.

We remark that "privacy" of values has been crucially used in "full" indifferentiability proofs for iterated random function [DRST12] and 3-round EMKD$_3$ cipher [GL16a]. In all, below we propose simulator strategy for EMSP$[\varphi]_4$ using this idea.

*Simulator for EMSP*$[\varphi]_4$*.* A query record $(x, y)$ is *public*, if either $P(x)$ or $P^{-1}(y)$ has been explicitly queried by $D$; otherwise, it is *internal*. In some sense, our simulator $\mathcal{S}$ detects all "public 2-chains", i.e., pairs of records $\big((x,y),(x',y')\big)$ (including the case $x = x'$) such that both $(x, y)$ and $(x', y')$ are "public". Such new "public 2-chains" arise in two cases:

(i) When $D$ issues a new query $P(x)$ or $P^{-1}(y)$ that is never encountered before;

(ii) When $D$ issues a query $P(x)$ or $P^{-1}(y)$ such that the corresponding record $(x, y)$ was internally created by previous simulator actions. This reflects $D$'s attempt of inspecting unknown internal information via queries.

For each such "2-chain" $\big((x,y),(x',y')\big)$, our simulator $\mathcal{S}$ views it as the 2nd and 3rd round evaluations of a chain, and tries to complete a "4-chain" $\big((x_1,y_1),(x,y),(x',y'),(x_4,y_4)\big)$ such that $E(k, k \oplus x_1) = \varphi^4(k) \oplus y_4$ (where $k = \varphi^{-2}(y \oplus x')$). The concrete simulator actions for completing this chain depends on whether records of the form $(\star, y_1)$ and $(x_4, \star)$ have been created before, and we refer to Sect. 3 for a detailed overview.

The records $(x_1, y_1)$ and $(x_4, y_4)$, if newly created, give rise to new "2-chains", e.g., $\big((x_1,y_1),(x,y)\big)$. But $(x_1, y_1)$ and $(x_4, y_4)$ (if new) are viewed as *internal and private*, and $\mathcal{S}$ does *not* complete these new "2-chains" immediately. Therefore, our simulator is quite simple and has *no recursion* at all. By this, the simulator complexity is obviously $O(q^2)$.

The crux is to prove that the "private" unqueried records won't cause trouble. Formally, we prove that when $D$ makes each query, every internal record has $O(2^n)$ possibilities for its "private" endpoint, even conditioned on *all the other* already created query records. Therefore, it is also uniform in the view of $D$. We prove this by considering parallel executions (of the interaction between $\mathcal{S}$ and $D$) with carefully chosen randomness, such that they produce the same records except for the "private" endpoint of exactly 1 internal record. These executions proceed with almost the same actions and provide the same information to $D$. Thus, $D$ cannot distinguish. We refer to Lemma 4 (Sect. 6) for details. Interestingly, it turns out crucial that the simulator does not handle the internal records at once. Otherwise, it will create a pile of "private" values with complicated dependency, and we cannot hope the private "endpoint" to be uniform conditioned on the other records.

*Further intuitions: An example.* To further understand our strategy, consider $D$ querying $P(x_i) \to y_i$, $i = 1, 2, 3, 4$, $x_{i+1} = y_i \oplus \varphi^i(k)$ again. The interaction consists of 4 steps.

Upon the initial query $P(x_1)$, $\mathcal{S}$ samples $y_1 \xleftarrow{\$} \{0,1\}^n$, creates $(x_1, y_1)$, detects the "2-chain" $\big((x_1,y_1),(x_1,y_1)\big)$ and completes the chain $\big((x'_0,y'_0),(x_1,y_1),(x_1,y_1),(x'_4,y'_4)\big)$ by creating two "internal" records $(x'_0, y'_0)$ and $(x'_4, y'_4)$. The new "2-chains" formed by $(x'_0, y'_0)$ and $(x'_4, y'_4)$ (e.g., $\big((x_1,y_1),(x'_0,y'_0)\big)$) won't be completed right now.

Then, $D$ makes the 2nd query $P(x_2)$. We assume that $D$ chooses $x_2 \neq x_1, x_4'$. On the other hand, as discussed, $D$ is unlikely to guess $x_2 = x_0'$. Therefore, $\mathcal{S}$ samples $y_2 \xleftarrow{\$} \{0,1\}^n \backslash \{y_1, y_0', y_4'\}$, creates $(x_2, y_2)$, and completes three "public 2-chains" $\big((x_1, y_1), (x_2, y_2)\big)$, $\big((x_2, y_2), (x_1, y_1)\big)$ and $\big((x_2, y_2), (x_2, y_2)\big)$ to $\big((x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)\big)$ and $\big((x_0'', y_0''), (x_2, y_2), (x_1, y_1), (x_4'', y_4'')\big)$ and $\big((x_0''', y_0'''), (x_2, y_2), (x_2, y_2), (x_4''', y_4''')\big)$.

Then, upon $D$ issuing the 3rd query $P(x_3)$, the record $(x_3, y_3)$ has been created at the end of the "4-chain" $\big((x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)\big)$. Similarly to discussed before, the record $(x_3, y_3)$ is "internal", with $x_3$ derivable from adversary known values and $y_3$ private and "unknown". $\mathcal{S}$ remarks $(x_3, y_3)$ as "public", and addresses the "public 2-chains" formed by $(x_3, y_3)$, including $\big((x_2, y_2), (x_3, y_3)\big)$ that has appeared but was shelved. For $\big((x_2, y_2), (x_3, y_3)\big)$, the corresponding "1st round" query record $(x_1, y_1)$ has been created, and $\mathcal{S}$ just needs to create an adapted record $(x_4, y_4)$ with $y_4 = \varphi^4(k) \oplus E(k, y_0)$.

This examples shows that *there remain some slide properties*: the two "4-chains" $\big((x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)\big)$ and $\big((x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\big)$ share 3 permutation evaluations. Though, our idea suffices in simulating two consistent "4-chains".

Finally, upon $D$ making the final query $P(x_4)$, $\mathcal{S}$ remarks the record $(x_4, y_4)$ as "public" and addresses the "public 2-chains" formed by $(x_4, y_4)$, including $\big((x_3, y_3), (x_4, y_4)\big)$ that was shelved. For $\big((x_3, y_3), (x_4, y_4)\big)$, the corresponding "1st round" query record $(x_2, y_2)$ has been created, and $\mathcal{S}$ just needs to create an adapted record $(x_5, y_5)$ with $y_5 = \varphi^5(k) \oplus E(\varphi(k), y_1)$, to have a "4-chain" $\big((x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5)\big)$. For traceability, we have to omit many "uninteresting" chain completions, but the above has exhibited an example of simulator execution, producing "4-chains" that are consistent with the ideal cipher $E$ *and* the limited slide property as well.

**Organization.** Sect. 2 serves notations and definitions. The remaining sections present the various steps of the proof: Sect. 3 presents our simulator; Sect. 4 introduces an intermediate system; Sect. 5 bounds simulator complexity; Sect. 6 formalizes privacy of internal values; Sect. 7 bounds the probability of certain "bad events" during simulations; Sect. 8 concludes on the failure probability of simulation; and finally Sect. 9 establishes indistinguishability of the real and ideal systems to complete the proof.

## 2 Preliminaries

Fix an integer $n$. An $n$-bit random permutation $\mathbf{p} : \{0,1\}^n \to \{0,1\}^n$ is a permutation that is uniformly chosen from all $(2^n)!$ possible choices, and its inverse is denoted by $\mathbf{p}^{-1}$. An ideal blockcipher $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is chosen randomly from the set of all blockciphers with key space $\{0,1\}^n$ and message and ciphertext space $\{0,1\}^n$. For each key $k \in \{0,1\}^n$, the map $E(k, \cdot)$ is a random permutation with inversion oracle $E^{-1}(k, \cdot)$.

The notion of sequential indifferentiability (seq-indifferentiability), introduced by Mandal et al. [MPS12], is a weakened variant of (full) indifferentiability of Maurer et al. [MRH04] tailored to a class of restricted distinguishers named *sequential distinguishers.* For concreteness, our formalism concentrates on blockciphers. Consider the blockcipher construction $\mathcal{C}^{\mathbf{p}}$ built upon several random permutations $\mathbf{p}$. A distinguisher $D^{\mathcal{C}^{\mathbf{p}}, \mathbf{p}}$ with oracle access to both the cipher and the underlying permutations is trying to distinguish $\mathcal{C}^{\mathbf{p}}$ from the ideal cipher $E$. Then, $D$ is *sequential*, if it proceeds in the following steps in a strict order: (1) queries the underlying permutation $\mathbf{p}$ in arbitrary direction; (2) queries the cipher $\mathcal{C}^{\mathbf{p}}$ in arbitrary; (3) outputs, and cannot query $\mathbf{p}$ again in this phase.

In this setting, if there is a simulator $\mathcal{S}^E$ that has access to $E$ and can mimic $\mathbf{p}$ such that in the view of any sequential distinguisher $D$, the system $(E, \mathcal{S}^E)$ is indistinguishable from $(\mathcal{C}^{\mathbf{p}}, \mathbf{p})$, then $\mathcal{C}^{\mathbf{p}}$ is *sequentially indifferentiable* (seq-indifferentiable) from $E$.

To characterize the adversarial power, we define a notion *total oracle query cost* of $D$, which refers to the total number of queries received by $\mathbf{p}$ (from $D$ or $\mathcal{C}^{\mathbf{p}}$) when $D$ interacts with $(\mathcal{C}^{\mathbf{p}}, \mathbf{p})$ [MPS12]. Then, seq-indifferentiability [MPS12, CS15] is defined as follows.

**Definition 1.** A blockcipher $\mathcal{C}^{\mathbf{p}}$ with oracle access to a random permutation $\mathbf{p}$ is statistically and strongly $(q, \sigma, t, \varepsilon)$-seq-indifferentiable from an ideal cipher $E$, if there exists a simulator $S^E$ such that for any sequential distinguisher $D$ of total oracle query cost $q$, $S^E$ issues at most $\sigma$ queries to $E$ and runs in time at most $t$, and

$$\left|\Pr_{\mathbf{p}}[D^{\mathcal{C}^{\mathbf{p}},\mathbf{p}} = 1] - \Pr_E[D^{E,\mathcal{S}^E} = 1]\right| \leq \varepsilon.$$

# 3 Simulator of EMSP$[\varphi]_4$

**Setup.** Our simulator $\mathcal{S}$ implements two public procedures $P$ and $P^{-1}$ as interfaces to the distinguisher for querying the random permutation $\mathbf{p}$ and its inverse $\mathbf{p}^{-1}$. Following previous works [CS15], our simulator $\mathcal{S}$ takes a random permutation $\mathbf{p}$ as *explicit randomness* to describe lazy sampling. Whenever $\mathcal{S}$ needs to sample an $n$-bit random value to respond $P(x)$ (resp., $P^{-1}(y)$), it queries $\mathbf{p}(x)$ (resp., $\mathbf{p}^{-1}(y)$) to "download" the response. Denote by $\mathcal{S}^{E,\mathbf{p}}$ the simulator that queries $E$ and $\mathbf{p}$.

**Permutation Query Records.** To distinguish between *internally created private permutation query records* and *public, adversarially chosen permutation queries*, $\mathcal{S}$ maintains two sets $\Pi_{pub}$ and $\Pi_{in}$, where the subscript *pub* stands for *public* and *in* stands for *internal*. Records in $\Pi_{in}$ are called *internal*, while records in $\Pi_{pub}$ are called *public*. Define $\Pi_{all} := \Pi_{pub} \cup \Pi_{in}$ as the union. Elements in the sets are 4-tuples of the form $(x, y, dir, num) \in \{0,1\}^n \times \{0,1\}^n \times \{\rightarrow, \leftarrow, \perp_{\leftarrow}, \perp_{\rightarrow}\} \times \mathbb{N}^+$, which are called (query) records. In each record $(x, y, dir, num) \in \Pi_{all}$, the first and second coordinates indicate the simulated relation $P(x) = y$, while the third coordinate $dir$ is the "direction" of the corresponding query and the fourth $num$ is the value of the query counter before creating this (query) record—our simulator $\mathcal{S}$ follows [ABD+13] and maintains a global query counter $qnum$ (initialized to 0 and increased right after $\mathcal{S}$ adding a new record to $\Pi_{all}$) to keep track of the order of creating query records. Concretely, $dir =\rightarrow$ or $\leftarrow$ if the record $(x, y, dir, num)$ has either $x = \mathbf{p}^{-1}(y)$ or $y = \mathbf{p}(x)$ from $\mathbf{p}$. In other cases, i.e., $dir = \perp_{\rightarrow}$ or $\perp_{\leftarrow}$, the record is called *adapted*. The detailed rules will be elaborated later. We sometimes simply write $(x, y, dir)$ or $(x, y)$, when the last coordinates are not of interest to the context.

*Private records $\Pi_{in}$ and procedure $InP$.* To distinguish internal permutation evaluations from adversarial queries, $\mathcal{S}$ implements a pair of *private* procedures $InP$ and $InP^{-1}$. Whenever $\mathcal{S}$ needs to internally acquire the value of $P(x)$, it calls $InP(x)$ (rather than $P(x)$). Now, if the corresponding record $(x, y, dir, num)$ has been in either $\Pi_{pub}$ or $\Pi_{in}$, $InP(x)$ simply returns $y$; otherwise, $InP(x)$ "downloads" the random response $y \leftarrow \mathbf{p}(x)$ and adds a new record $(x, y, \rightarrow, qnum)$ to $\Pi_{in}$ and then returns $y$. As discussed in Sect. 1.1, the intuition is the value $y$ is kept secret in the state of $\mathcal{S}$ and is "unknown" to the distinguisher. A call to $InP^{-1}(y)$ runs similar by symmetry.

Whenever $\mathcal{S}$ is to adapt, it adds a new record $(x, y, dir, qnum)$ to the internal set $\Pi_{in}$, where $dir = \perp_{\leftarrow}$ or $\perp_{\rightarrow}$. The intuition, which is also found in Sect. 1.1, is that every adapted record $(x, y, dir, qnum)$ has either $x$ or $y$ derived from a new ideal cipher query response that happens right before $\mathcal{S}$ creating it. We refer to the description of $ProcessRecord(x, y, dir)$ below for example.

*Public records $\Pi_{pub}$.* Finally, the set $\Pi_{pub}$ keeps all the records that have been (explicitly) queried by the distinguisher. This includes both the records newly created "straightforwardly" for adversarial queries and the records that were moved from $\Pi_{in}$ to $\Pi_{pub}$ due to adversarial queries (as will be elaborated later).

Additional notations for $\Pi_{all}$. $\mathcal{S}$ will ensure that: (i) the union $\Pi_{all} := \Pi_{pub} \cup \Pi_{in}$ always defines a partial permutation; (ii) the sets $\Pi_{pub}$ and $\Pi_{in}$ are always disjoint. $\mathcal{S}$ *aborts* whenever it cannot ensure such consistency anymore (thus, a major part of our proof is devoted to show that abortions are unlikely). By this, for $i \in \{pub,\ in\}$, we denote by $domain(\Pi_i)$ ($range(\Pi_i)$, resp.) the (time-dependent) set of all $n$-bit values $x$ ($y$, resp.) satisfying $\exists z \in \{0,1\}^n$ s.t. $(x,z) \in \Pi_i$ ($(z,y) \in \Pi_i$, resp.). We further denote by $\Pi_i(x)$ ($\Pi_i^{-1}(y)$, resp.) the corresponding value of $z$.

**Chains and *CompletedChains*.** With the sets $\Pi_{all}$ introduced before, we now define *2-chain*, *3-chain* and *4-chains* to ease describing simulation strategy.

**Definition 2.** A *2-chain* is an ordered pair of records $\big((x,y),(x',y')\big) \in (\Pi_{all})^2$. A 2-chain $\big((x,y),(x',y')\big)$ is *public*, if both $(x,y)$ and $(x',y')$ are in $\Pi_{pub}$.

A *3-chain* is an ordered triple of records $\big((x,y),(x',y'),(x'',y'')\big) \in (\Pi_{all})^3$ such that $y' \oplus x'' = \varphi(y \oplus x')$.

A *4-chain* is a 4-tuple $\big((x_1,y_1),(x_2,y_2),(x_3,y_3),(x_4,y_4)\big) \in (\Pi_{all})^4$ such that $y_2 \oplus x_3 = \varphi(y_1 \oplus x_2)$, $y_3 \oplus x_4 = \varphi^2(y_2 \oplus x_1)$, and $E\big(k, k \oplus x_1\big) = \varphi^4(k) \oplus y_4$ for $k = \varphi^{-1}(y_1 \oplus x_2)$.

A 2-chain $\big((x,y),(x',y')\big) \in (\Pi_{all})^2$ is *in a (completed) 4-chain*, if there exists a 4-chain $\big((x'',y''),(x,y),(x',y'),(x''',y''')\big)$ for some $(x'',y''),(x''',y''') \in \Pi_{all}$. This means $\mathcal{S}$ has performed chain completion for $\big((x,y),(x',y')\big)$. To keep track, $\mathcal{S}$ maintains a set *CompletedChains* with elements $\big((x,y),(x',y')\big) \in (\Pi_{all})^2$ for such "processed" 2-chain. Similarly, a 3-chain $\big((x,y),(x',y'),(x'',y'')\big) \in (\Pi_{all})^3$ is *in a 4-chain*, if there exists a 4-chain $\big((x,y),(x',y'),(x'',y''),(x''',y''')\big)$ or $\big((x''',y'''),(x,y),(x',y'),(x'',y'')\big)$ for some $(x''',y''') \in \Pi_{all}$.

**Simulator Actions.** As mentioned in the Introduction, our simulator $\mathcal{S}$ performs chain completion actions when $D$ issues a query that is either new or corresponds to an internal query record. In both cases, $\mathcal{S}$ addresses all newly formed *public* 2-chains.

In detail, upon $D$ querying $P(x)$, $\mathcal{S}$ checks $\Pi_i$, $i \in \{pub, in\}$, to see whether the corresponding record $(x,y)$ has been created. We distinguish three cases.

- Case 1: $x \in domain(\Pi_{pub})$. In this case, $\mathcal{S}$ simply returns $\Pi_{pub}(x)$.
- Case 2: $x \in domain(\Pi_{in})$. In this case, $\mathcal{S}$ moves the record $(x,y,dir,num)$ from $\Pi_{in}$ to $\Pi_{pub}$. Then, $\mathcal{S}$ makes a call to a private procedure $ProcessRecord(x,y,dir)$ to complete chains (see below for its detailed actions). After these, $\mathcal{S}$ returns $y$ to $D$;
- Case 3: $x \notin domain(\Pi_{all})$. In this case, $\mathcal{S}$ queries $y \leftarrow \mathbf{p}(x)$ for $y$ and adds a record $(x,y,\rightarrow,qnum)$ to $\Pi_{pub}$. Then, $\mathcal{S}$ calls $ProcessRecord(x,y,\rightarrow)$ to complete chains (see below). After these, $\mathcal{S}$ returns $y$ to $D$.

Therefore, the two sets $\Pi_{pub}$ and $\Pi_{in}$ are always disjoint.

$ProcessRecord(x,y,dir)$. As intuition, $D$ is unlikely to guesses $x$ for an internal record $(x,y,dir) \in \Pi_{in}$ with $dir \in \{\leftarrow, \bot_{\leftarrow}\}$. Therefore, in $ProcessRecord(x,y,dir)$ which is called due to $D$ querying $P(x)$, it is expected to have $dir \in \{\rightarrow, \bot_{\rightarrow}\}$. In this case, $\mathcal{S}$ checks relevant partial chains as follows.

First, if there exists a 2-chain $\big((x'',y''),(x',y')\big)$ such that $(x',y') \in \Pi_{pub}$ and $x = y' \oplus \varphi(y'' \oplus x')$, then $\mathcal{S}$ recognizing a 3-chain $\big((x'',y''),(x',y'),(x,y)\big)$. $\mathcal{S}$ then computes $k \leftarrow \varphi^{-1}(y'' \oplus x')$, $u \leftarrow x'' \oplus k$, $v \leftarrow E(k,u)$, $y_4 \leftarrow v \oplus \varphi^4(k)$, and $x_4 \leftarrow y \oplus \varphi^3(k)$. $\mathcal{S}$ then adds the adapted record $(x_4, y_4, \bot_{\rightarrow}, qnum)$ to $\Pi_{in}$ to complete a 4-chain $\big((x'',y''),(x',y'),(x,y),(x_4,y_4)\big)$. The adapted record has $dir = \bot_{\rightarrow}$, meaning that $y_4$ is derived from the new ideal cipher query response $v \leftarrow E(k,u)$.

Then, $(x,y)$ triggers detecting two types of public 2-chains $\big((x',y'),(x,y)\big)$ (including the case of $x' = x$) and $\big((x,y),(x',y')\big)$ ($x' \neq x$). $\mathcal{S}$ completes them as follows.

1. First, for each public 2-chain $\big((x', y'), (x, y)\big)$, $\mathcal{S}$ computes $k \leftarrow \varphi^{-2}(y' \oplus x)$, $x_4 \leftarrow y \oplus \varphi^3(k)$, $y_4 \leftarrow InP(x_4)$, $v \leftarrow y_4 \oplus \varphi^4(k)$, $u \leftarrow E^{-1}(k, v)$, $x_1 \leftarrow u \oplus k$, $y_1 \leftarrow x' \oplus \varphi(k)$. $\mathcal{S}$ then adds the adapted record $(x_1, y_1, \perp_{\leftarrow})$ to $\Pi_{in}$ to complete a 4-chain $\big((x_1, y_1), (x', y'), (x, y), (x_4, y_4)\big)$. The adapted record has $dir = \perp_{\leftarrow}$, meaning that $x_1$ is derived from the new ideal cipher query response $u \leftarrow E^{-1}(k, v)$.
2. Then, for each public 2-chain $\big((x, y), (x', y')\big)$, $\mathcal{S}$ computes $k \leftarrow \varphi^{-2}(y \oplus x')$, $y_1 \leftarrow x \oplus \varphi(k)$, $x_1 \leftarrow InP^{-1}(y)$, $u \leftarrow x_1 \oplus k$, $v \leftarrow E(k, u)$, $y_4 \leftarrow v \oplus \varphi^4(k)$, and $x_4 \leftarrow y' \oplus \varphi^3(k)$. $\mathcal{S}$ then adds the adapted record $(x_4, y_4, \perp_{\rightarrow})$ to $\Pi_{in}$ to complete a 4-chain $\big((x_1, y_1), (x, y), (x', y'), (x_4, y_4)\big)$.

As discussed, $\mathcal{S}$ does *not* detect the (other) 2-chains and 3-chains formed by internal records. Thus, the above chain detection and completions are "one-shot deal".

Upon $D$ querying $P^{-1}(y)$ and inducing a call to $ProcessRecord(x, y, dir)$ with $dir \in \{\leftarrow, \perp_{\leftarrow}\}$, the actions of $\mathcal{S}$ are similar by symmetry. A pseudocode description of $\mathcal{S}$ is given as follows. Note that $\mathcal{S}$ maintains an additional set $ET$ to keep track of ideal cipher queries it has made. Later in Sect. 4, we will consider another simulator $\mathcal{T}$ modified from $\mathcal{S}$ by adding a number of lines. To this end, we put the added lines into boxes and mark them red to highlight. The reader can ignore these boxed statements at the moment.

```
Simulator $\mathcal{S}^{E,\mathbf{P}}$          $\boxed{\textbf{Simulator } \mathcal{T}^{E,\mathbf{P}}}$
Variables: Sets $\Pi_{pub}, \Pi_{in}, ET, CompletedChains$, all initially empty
           // $\Pi_{all} := \Pi_{pub} \cup \Pi_{in}$
           Integer $qnum$, initialized to 0

public procedure $P(x)$
    $\boxed{CheckPrivacy(x)}$
    if $x \in domain(\Pi_{pub})$ then
        return $\Pi_{pub}(x)$
    // It then holds $x \notin domain(\Pi_{pub})$
    if $x \notin domain(\Pi_{in})$ then
        $y \leftarrow InP(x)$
    // $x \in domain(\Pi_{in})$
    if $(x, y, \star, \star) \in \Pi_{in}$ then
        $\boxed{CheckInternalColl(x, y)}$
        $\Pi_{in} \leftarrow \Pi_{in} \setminus \{(x, y, dir, num)\}$
    $\Pi_{pub} \leftarrow \Pi_{pub} \cup \{(x, y, dir, num)\}$
    $\boxed{\textbf{for each } (x, y) \in (\Pi_{in}) \textbf{ do}}$
        $\boxed{CheckInterV3Chain(x, y)}$
    $ProcessRecord(x, y, dir)$
    return $\Pi_{pub}(x)$

public procedure $P^{-1}(y)$
    $\boxed{CheckPrivacy^{-1}(y)}$
    if $y \in range(\Pi_{pub})$ then
        return $\Pi_{pub}^{-1}(y)$
    // It then holds $y \notin range(\Pi_{pub})$
    if $y \notin range(\Pi_{in})$ then
        $x \leftarrow InP^{-1}(y)$
    // $y \in range(\Pi_{in})$
    if $(x, y, \star, \star) \in \Pi_{in}$ then
        $\boxed{CheckInternalColl^{-1}(x, y)}$
        $\Pi_{in} \leftarrow \Pi_{in} \setminus \{(x, y, dir, num)\}$
    $\Pi_{pub} \leftarrow \Pi_{pub} \cup \{(x, y, dir, num)\}$
    $\boxed{\textbf{for each } (x, y) \in (\Pi_{in}) \textbf{ do}}$
        $\boxed{CheckInterV3Chain(x, y)}$
    $ProcessRecord(x, y, dir)$
    return $\Pi_{pub}^{-1}(y)$

private procedure $ProcessRecord(x, y, dir)$
    $Check3Chains(x, y, dir)$
    $Check2Chains(x, y)$

$\boxed{\textbf{private procedure } CheckPrivacy(x)}$
    $\boxed{\textbf{if } \exists (x, y, dir) \in (\Pi_{in}) \text{ s.t. } dir \in \{\leftarrow, \perp_{\leftarrow}\} \textbf{ then abort}}$
    $\boxed{\textbf{if } \exists (x_1, y_1, dir_1) \in \Pi_{all},\ (x_2, y_2) \in \Pi_{all} \text{ s.t. } x = y_2 \oplus \varphi(y_1 \oplus x_2), \text{ and:}}$
            $\boxed{\text{(i) } (x_1, y_1, dir_1) \in (\Pi_{in}) \text{ and } dir_1 \in \{\rightarrow, \perp_{\rightarrow}\}; \text{ or}}$
            $\boxed{\text{(ii) } (x_2, y_2) \in (\Pi_{in})}$
        $\boxed{\textbf{then abort} \qquad // \text{ A 3-chain } \big((x_1, y_1), (x_2, y_2), (x, \star)\big)}$
    if $\exists (x_1, y_1), (x_2, y_2, dir_2), (x_3, y_3) \in \Pi_{all}$ s.t. $x_1 \oplus \varphi^{-1}(y_1 \oplus x_2) = x_3 \oplus \varphi^{-1}(y_3 \oplus x)$, and:
              (i) $(x_1, y_1) \in \Pi_{in}$; or
              (ii) $(x_2, y_2) \in \Pi_{in}$ and $dir_2 \in \{\leftarrow, \perp_{\leftarrow}\}$; or
              (iii) $(x_3, y_3) \in \Pi_{in}$
      // Two 2-chains $\big((x_1, y_1), (x_2, y_2)\big)$ and $\big((x_3, y_3), (x, \star)\big)$ collide at left
      then abort

private procedure $CheckPrivacy^{-1}(y)$
    if $\exists (x, y, dir) \in \Pi_{in}$ s.t. $dir \in \{\rightarrow, \perp_{\rightarrow}\}$ then abort
    if $\exists (x_2, y_2) \in \Pi_{all}$, $(x_3, y_3, dir_3) \in \Pi_{all}$ s.t. $y = x_2 \oplus \varphi^{-1}(y_2 \oplus x_3)$, and:
              (i) $(x_2, y_2) \in \Pi_{in}$; or
              (ii) $(x_3, y_3, dir_3) \in \Pi_{in}$ and $dir'' \in \{\leftarrow, \perp_{\leftarrow}\}$
      then abort
    if $\exists (x_1, y_1, dir_1), (x_2, y_2), (x_4, y_4) \in \Pi_{all}$ s.t. $y_2 \oplus \varphi(y_1 \oplus x_2) = y_4 \oplus \varphi(y \oplus x_4)$, and:
              (i) $(x_1, y_1) \in \Pi_{in}$ and $dir_1 \in \{\rightarrow, \perp_{\rightarrow}\}$; or
              (ii) $(x_2, y_2) \in \Pi_{in}$; or
              (iii) $(x_4, y_4) \in \Pi_{in}$
      then abort

private procedure $CheckInternalColl(x, y)$
    if $\exists (x_2, y_2), (x_3, y_3) \in \Pi_{all}$ s.t. $x_3 = y_2 \oplus \varphi(y \oplus x_2)$    // A 3-chain $\big((x, y), (x_2, y_2), (x_3, y_3)\big)$
      then abort
    if $\exists (x_2, y_2), (x_3, y_3), (x_4, y_4) \in \Pi_{all}$ s.t.: (i) $x \oplus \varphi^{-1}(y \oplus x_2) = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4)$; or
                                    (ii) $y_2 \oplus \varphi(y \oplus x_2) = y_4 \oplus \varphi(y_3 \oplus x_4)$
      // Two 2-chains $\big((x, y), (x_2, y_2)\big)$ and $\big((x_3, y_3), (x_4, y_4)\big)$ collide at either left or right
      then abort
    if $\exists (x_1, y_1), (x_3, y_3), (x_4, y_4) \in \Pi_{all}$ s.t. $y \oplus \varphi(y_1 \oplus x) = y_4 \oplus \varphi(y_3 \oplus x_4)$
      // Two 2-chains $\big((x_1, y_1), (x, y)\big)$ and $\big((x_3, y_3), (x_4, y_4)\big)$ collide at right
      then abort

private procedure $CheckInternalColl^{-1}(x, y)$
    if $\exists (x_1, y_1), (x_2, y_2) \in \Pi_{all}$ s.t. $x = y_2 \oplus \varphi(y_1 \oplus x_2)$    // A 3-chain $\big((x_1, y_1), (x_2, y_2), (x, y)\big)$
      then abort
    if $\exists (x_1, y_1), (x_3, y_3), (x_4, y_4) \in \Pi_{all}$ s.t.: (i) $x_1 \oplus \varphi^{-1}(y_1 \oplus x) = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4)$; or
                                    (ii) $y \oplus \varphi(y_1 \oplus x) = y_4 \oplus \varphi(y_3 \oplus x_4)$
      // Two 2-chains $\big((x_1, y_1), (x, y)\big)$ and $\big((x_3, y_3), (x_4, y_4)\big)$ collide at either left or right
      then abort
    if $\exists (x_2, y_2), (x_3, y_3), (x_4, y_4) \in \Pi_{all}$ s.t. $x \oplus \varphi^{-1}(y \oplus x_2) = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4)$
      // Two 2-chains $\big((x, y), (x_2, y_2)\big)$ and $\big((x_3, y_3), (x_4, y_4)\big)$ collide at left
```

```
        then abort
private procedure $CheckInterV3Chain(x, y)$
    if there exist distinct 2-chains $\big((x_1, y_1), (x_2, y_2)\big) \in (\Pi_{pub})^2$
            and $\big((x_3, y_3), (x_4, y_4)\big) \in (\Pi_{pub})^2$ s.t. $\varphi(y' \oplus x) = y \oplus x''$, where
                $y' = x_1 \oplus \varphi^{-1}(y_1 \oplus x_2)$ and $x'' = y_4 \oplus \varphi(y_3 \oplus x_4)$
        then abort            // Internally linking a “virtual” 3-chain

private procedure $InP(x)$
    if $x \notin domain(\Pi_{all})$ then
        $y \leftarrow \mathbf{p}(x)$
        if $y \in range(\Pi_{all})$ then abort
        $CheckRecord(x, y, \rightarrow, qnum)$
        $\Pi_{in} \leftarrow \Pi_{in} \cup \{(x, y, \rightarrow, qnum)\}$
        $qnum \leftarrow qnum + 1$
    return $y$

private procedure $InP^{-1}(y)$
    if $y \notin range(\Pi_{all})$ then
        $x \leftarrow \mathbf{p}^{-1}(y)$
        if $x \in domain(\Pi_{all})$ then abort
        $CheckRecord(x, y, \leftarrow, qnum)$
        $\Pi_{in} \leftarrow \Pi_{in} \cup \{(x, y, \leftarrow, qnum)\}$
        $qnum \leftarrow qnum + 1$
    return $x$

private procedure $Check3Chains(x, y, dir)$
    if $dir \in \{\rightarrow, \perp_{\rightarrow}\}$ then
        forall $(x', y') \in \Pi_{pub}$ do
            forall $(x'', y'') \in \Pi_{in}$ do    // 3-chains of the form $\big((x'', y''), (x', y'), (x, y)\big)$
                if $x = y' \oplus \varphi(y'' \oplus x')$ and $\big((x', y'), (x, y)\big) \notin CompletedChains$ then
                    $k \leftarrow \varphi^{-1}(y'' \oplus x'), Complete^{+}(y, k)$
    else if $dir \in \{\leftarrow, \perp_{\leftarrow}\}$ then
        forall $(x', y') \in \Pi_{pub}$ do
            forall $(x'', y'') \in \Pi_{in}$ do    // 3-chains of the form $\big((x, y), (x', y'), (x'', y'')\big)$
                if $y = x' \oplus \varphi^{-1}(x' \oplus y'')$ and $\big((x, y), (x', y')\big) \notin CompletedChains$ then
                    $k \leftarrow \varphi^{-2}(y \oplus x'), Complete^{-}(x, k)$

private procedure $Check2Chains(x, y)$
    forall $(x', y') \in \Pi_{pub}$ do              // 2-chains of the form $\big((x', y'), (x, y)\big)$
        if $\big((x', y'), (x, y)\big) \notin CompletedChains$ then
            $Complete^{-}(x', \varphi^{-2}(x \oplus y'))$
    forall $(x', y') \in \Pi_{pub} \backslash \{(x, y)\}$ do      // 2-chains of the form $\big((x, y), (x', y')\big)$
        if $\big((x, y), (x', y')\big) \notin CompletedChains$ then
            $Complete^{+}(y', \varphi^{-2}(x' \oplus y))$

private procedure $Complete^{+}(y_3, k)$
    $x_4 \leftarrow y_3 \oplus \varphi^3(k)$, $x_3 \leftarrow InP^{-1}(y_3)$
    $y_2 \leftarrow x_3 \oplus \varphi^2(k)$, $x_2 \leftarrow InP^{-1}(y_2)$
    $y_1 \leftarrow x_2 \oplus \varphi(k)$, $x_1 \leftarrow InP^{-1}(y_1)$
    $u \leftarrow x_1 \oplus k$, $v \leftarrow E(k, u)$
    $ET \leftarrow ET \cup \{(k, u, v)\}$, $y_4 \leftarrow v \oplus \varphi^4(k)$
    $CheckRecord(x_4, y_4, \perp_{\rightarrow}, qnum)$
    $Adapt(x_4, y_4, \perp_{\rightarrow}, qnum)$
    $qnum \leftarrow qnum + 1$

private procedure $Complete^{-}(x_2, k)$
    $y_1 \leftarrow x_2 \oplus \varphi(k)$, $y_2 \leftarrow InP(x_2)$
    $x_3 \leftarrow y_2 \oplus \varphi^2(k)$, $y_3 \leftarrow InP(x_3)$
    $x_4 \leftarrow y_3 \oplus \varphi^3(k)$, $y_4 \leftarrow InP(x_4)$
    $v \leftarrow y_4 \oplus \varphi^4(k)$, $u \leftarrow E^{-1}(k, v)$
    $ET \leftarrow ET \cup \{(k, u, v)\}$, $x_1 \leftarrow u \oplus k$
    $CheckRecord(x_1, y_1, \perp_{\leftarrow}, qnum)$
    $Adapt(x_1, y_1, \perp_{\leftarrow}, qnum)$
    $qnum \leftarrow qnum + 1$

private procedure $Adapt(x, y, dir, num)$
    if $x \in domain(\Pi_{all})$ or $y \in range(\Pi_{all})$ then abort
    $\Pi_{in} \leftarrow \Pi_{in} \cup \{(x, y, dir, num)\}$

private procedure $CheckRecord(x, y, dir, num)$
```

```
if $\exists (x', y') \in \Pi_{all}$ s.t. $y \oplus \varphi(x) = y' \oplus \varphi(x')$ then abort
$\Pi^*_{all} \leftarrow \Pi_{all} \cup \{(x, y, dir, num)\}$                // A temporary record set
if there exist distinct 2-chains $\big((x_1, y_1), (x_2, y_2)\big) \in (\Pi_{all})^2$
        and $\big((x_3, y_3), (x_4, y_4)\big) \in (\Pi_{all})^2$ s.t. $\varphi(y' \oplus x) = y \oplus x''$, where
                $y' = x_1 \oplus \varphi^{-1}(y_1 \oplus x_2)$ and $x'' = y_4 \oplus \varphi(y_3 \oplus x_4)$
    then abort            // Linking a “virtual” 3-chain
if $\exists (x_1, y_1, dir_1, num_1), (x_2, y_2, dir_2, num_2), (x_3, y_3, dir_3, num_3) \in (\Pi^*_{all})^3$
        s.t. $y_2 \oplus \varphi(y_1 \oplus x_2) = x_3$, and
        (i) $num_1 \geq num_2, num_3$, $dir_1 = \rightarrow$ or $\bot_{\rightarrow}$; or
        (ii) $num_2 \geq num_1, num_3$; or
        (iii) $num_3 \geq num_1, num_2$, $dir_3 = \leftarrow$ or $\bot_{\leftarrow}$,
    then abort            // Unexpected 3-chains
if there exist distinct 2-chains $\big((x_1, y_1, dir_1, num_1), (x_2, y_2, dir_2, num_2)\big) \in (\Pi^*_{all})^2$
        and $\big((x_3, y_3, dir_3, num_3), (x_4, y_4, dir_4, num_4)\big) \in (\Pi^*_{all})^2$
        s.t. $y_2 \oplus \varphi(y_1 \oplus x_2) = y_4 \oplus \varphi(y_3 \oplus x_4)$, and
        (i) $num_1 \geq num_2, num_3, num_4$, $dir = \rightarrow$ or $\bot_{\rightarrow}$; or
        (ii) $num_2 \geq num_1, num_3, num_4$.
    then abort            // Right endpoints of two 2-chains collide
if there exist distinct 2-chains $\big((x_1, y_1, dir_1, num_1), (x_2, y_2, dir_2, num_2)\big) \in (\Pi^*_{all})^2$
        and $\big((x_3, y_3, dir_3, num_3), (x_4, y_4, dir_4, num_4)\big) \in (\Pi^*_{all})^2$
        s.t. $x_1 \oplus \varphi^{-1}(y_1 \oplus x_2) = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4)$, and
        (i) $num_1 \geq num_2, num_3, num_4$; or
        (ii) $num_2 \geq num_1, num_3, num_4$, $dir = \leftarrow$ or $\bot_{\leftarrow}$.
    then abort            // Left endpoints of two 2-chains collide
$\Pi_{all} \leftarrow \Pi_{all} \cup \{(x, y, dir, num)\}$        // Add the record to $\Pi_{all}$
```

## 4 Intermediate System $\Sigma_2$ and Its Basic Properties

We follow [MPS12, CS15] and use three systems $\Sigma_1(E, \mathcal{S}^{E,\mathbf{p}})$, $\Sigma_2(\mathrm{EMSP}[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}}, \mathcal{T}^{E,\mathbf{p}})$ and $\Sigma_3(\mathrm{EMSP}[\varphi]_4^{\mathbf{p}}, \mathbf{p})$ for the proof. The system $\Sigma_1$ captures the ideal world $(E, \mathcal{S}^{E,\mathbf{p}})$, while $\Sigma_3$ captures the real world $(\mathrm{EMSP}[\varphi]_4^{\mathbf{p}}, \mathbf{p})$.

The intermediate system $\Sigma_2$ uses another simulator $\mathcal{T}^{E,\mathbf{p}}$ that is modified from $\mathcal{S}^{E,\mathbf{p}}$ by adding a number of abortions upon certain bad events. Briefly, the events cover the appearance of *bad queries structures* or *collisions in private values* due to collisions among random values. In the pseudocode in Sect. 3, six private procedures *CheckRecord*, *CheckPrivacy*, *CheckPrivacy*$^{-1}$, *CheckInternalColl*, *CheckInternalColl*$^{-1}$ and *CheckInterV3Chain* are used to check if such conditions are fulfilled. In addition, the blockcipher oracle in $\Sigma_2$ is instantiated with the $\mathrm{EMSP}[\varphi]_4$ construction that queries $\mathcal{T}^{E,\mathbf{p}}$ for computations.

For the remaining of the proof, we consider a fixed, deterministic sequential distinguisher $D$ that has total oracle query cost $q$. The central part is to analyze the intermediate system $\Sigma_2$, and to establish two claims: (a) the simulator $\mathcal{T}^{E,\mathbf{P}}$ has bounded complexity; (b) the real and ideal worlds are indistinguishable.

We start by exhibiting some basic properties of $\Sigma_2$ executions. An execution of the game $\Sigma_2$ is *good*, if it does not abort. Otherwise, it is *bad*. A *simulator cycle* consists of the execution period starting from when the distinguisher $D$ makes a query to when $D$ receives an answer (which may be an abort message). We distinguish three types of simulator cycles as follows.

(i) $D$ queries $P(x)$ such that $x \notin domain(\Pi_{all})$; or, symmetrically, $D$ queries $P^{-1}(y)$ such that $y \notin range(\Pi_{all})$. We call them **new simulator cycles**.

(ii) $D$ queries $P(x)$ such that $x \in domain(\Pi_{in})$; or, symmetrically, $D$ queries $P^{-1}(y)$ such that $y \in range(\Pi_{in})$. We call them **transferring cycles**.

(iii) $D$ queries $P(x)$ or $P^{-1}(y)$ such that $x \in domain(\Pi_{pub})$ or $y \in range(\Pi_{pub})$. This is the trivial case, and $\mathcal{S}$ simply replies with the corresponding values in $\Pi_{pub}$.

**Invariants.** Since a new record $(x, y, dir, num)$ is added to the sets only if it has passed a series of checks, various "good invariants" about the record data structure can be shown to hold unconditionally at any point in any $\Sigma_2$ execution. We list them as follows.

**Inv1***: No "cycle" within two records.* Due to line 121, there do not exist two distinct records $(x, y), (x', y') \in (\Pi_{all})^2$ such that $y \oplus \varphi(x) = y' \oplus \varphi(x')$.

**Inv2***: No unexpected 3-chains.* Due to line 126, there do not exist three (not necessarily distinct) records $(x_1, y_1, dir_1, num_1), (x_2, y_2, dir_2, num_2), (x_3, y_3, dir_3, num_3) \in (\Pi_{all})^3$ such that $y_2 \oplus \varphi(y_1 \oplus x_2) = x_3$, and

- $num_1 \geq num_2, num_3$, $dir_1 = \rightarrow$ or $\perp_\rightarrow$; or
- $num_2 \geq num_1, num_3$; or
- $num_3 \geq num_1, num_2$, $dir_3 = \leftarrow$ or $\perp_\leftarrow$,

**Inv3***: No unexpected "right collision" among two 2-chains.* Due to line 128, there do not exist two distinct public 2-chains $\big((x_1, y_1, dir_1, num_1), (x_2, y_2, dir_2, num_2)\big)$ and $\big((x_3, y_3, dir_3, num_3), (x_4, y_4, dir_4, num_4)\big)$ such that $y_2 \oplus \varphi(y_1 \oplus x_2) = y_4 \oplus \varphi(y_3 \oplus x_4)$, and

- $num_1 > num_2, num_3, num_4$, $dir = \rightarrow$ or $\perp_\rightarrow$; or
- $num_2 > num_1, num_3, num_4$.

**Inv4***: No unexpected "left collision" among two 2-chains.* Due to line 130, there do not exist two distinct public 2-chains $\big((x_1, y_1, dir_1, num_1), (x_2, y_2, dir_2, num_2)\big)$ and $\big((x_3, y_3, dir_3, num_3), (x_4, y_4, dir_4, num_4)\big)$ such that $x_1 \oplus \varphi^{-1}(y_1 \oplus x_2) = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4)$, and

- $num_1 > num_2, num_3, num_4$; or
- $num_2 > num_1, num_3, num_4$, $dir = \leftarrow$ or $\perp_\leftarrow$.

**Basic Properties of Simulation.** First, by inspecting the simulator description in Sect. 3, we make the following observation on 4-chains.

**Proposition 1.** *Assume that the simulator $\mathcal{T}^{E,\mathbf{P}}$ makes a call to Complete$^+$/Complete$^-$, which completes a 4-chain $\big((x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\big)$. Then, the current simulator cycle was necessarily due to $D$ querying $P(x_2), P^{-1}(y_2), P(x_3)$ or $P^{-1}(y_3)$. In addition, if this call does not abort, then $(x_2, y_2), (x_3, y_3) \in \Pi_{pub}$ after this call returns.*

Besides, every internal record $(x, y, dir)$ has either $x$ or $y$ "adjacent" to a public 2-chain.

**Proposition 2.** *Assume that in a simulator cycle due to $D$ querying $P(x^*) \to y^*$ or $P^{-1}(y^*) \to x^*$, $\mathcal{T}$ adds a record $(x, y, dir, num)$ to $\Pi_{in}$. Then, by our design of $\mathcal{T}$, $(x, y, dir, num)$ necessarily falls into either of the following two cases:*

- *Case 1: $dir \in \{\to, \bot_\to\}$. In this case, there exists $(x', y') \in \Pi_{pub}$ such that either of the following holds:*
  - *$x = y' \oplus \varphi(y^* \oplus x')$, i.e., $x$ is the "right endpoint" of the 2-chain $\big((x^*, y^*), (x', y')\big)$;*
  - *$x = y^* \oplus \varphi(y' \oplus x^*)$, i.e., $x$ is the "right endpoint" of the 2-chain $\big((x', y'), (x^*, y^*)\big)$.*
- *Case 2: $dir \in \{\leftarrow, \bot_\leftarrow\}$. In this case, there exists $(x', y') \in \Pi_{pub}$ such that either of the following holds:*
  - *$y = x^* \oplus \varphi^{-1}(y^* \oplus x')$, i.e., $y$ is the "left endpoint" of the 2-chain $\big((x^*, y^*), (x', y')\big)$;*
  - *$y = x' \oplus \varphi^{-1}(y' \oplus x^*)$, i.e., $y$ is the "left endpoint" of the 2-chain $\big((x', y'), (x^*, y^*)\big)$.*

**Lemma 1.** *At the end of a simulator cycle, if $\mathcal{T}^{E,\mathbf{P}}$ did not abort, then every public 2-chain is in a corresponding 4-chain.*

*Proof.* At the beginning, the claim holds. By induction, assume that it holds before a non-trivial cycle. Then, the cycle adds exactly 1 record $(x, y)$ to $\Pi_{pub}$. By construction, all new public 2-chains due to $(x, y)$ are completed to 4-chains in this cycle. □

A 3-chain $\big((x, y), (x', y'), (x'', y'')\big)$ is *bad*, if $(x', y') \in \Pi_{in}$, i.e., the "middle" record is internal. Similarly, a 4-chain $\big((x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\big)$ is *bad*, if either $(x_2, y_2)$ or $(x_3, y_3)$ is in $\Pi_{in}$. Below we prove 3-chain and 4-chain are always good.

**Lemma 2.** *At any time during a non-aborting $\Sigma_2$ execution, all 3-chains are good.*

*Proof.* Assume that a bad 3-chain $\big((x, y, dir, num), (x', y', dir', num'), (x'', y'', dir'', num'')\big)$ with $(x', y') \in \Pi_{in}$ appears at some time. It cannot be $num' \geq num, num''$: otherwise, it contradicts **Inv2** regardless of the value of $dir'$. This means either $num > num' \wedge num \geq num''$ or $num'' > num' \wedge num'' \geq num$. Though, it cannot be $num > num' \wedge num = num''$: otherwise, the 3-chain $\big((x, y, dir, num), (x', y', dir', num'), (x, y, dir, num)\big)$ contradicts **Inv2**. Similarly, it cannot be $num'' > num' \wedge num'' = num$.

Thus, wlog assume $num > num' \wedge num > num''$ for the remaining, i.e., $\mathcal{T}^{E,\mathbf{P}}$ (roughly) first creates $(x', y'), (x'', y'')$, and then creates $(x, y)$. Then $dir \in \{\leftarrow, \bot_\leftarrow\}$, as otherwise it contradicts **Inv2** again. For convenience, we call the simulator cycle that creates $(x, y, dir)$ *the triggering cycle.* We exclude 5 possibilities as follows.

**Case 1: $(x, y, \leftarrow)$ is created due to $D$ querying $P^{-1}(y)$.** This is not possible: otherwise, it holds $y = x' \oplus \varphi^{-1}(y' \oplus x'')$, and $\mathcal{T}^{E,\mathbf{P}}$ would have aborted in the call to $CheckPrivacy^{-1}(y)$ (and would not create $(x, y, \leftarrow)$).

**Case 2: $(x, y, dir, num), (x', y', dir', num')$ and $(x'', y'', dir'', num'')$ are created in the same cycle.** As argued, it holds $dir \in \{\leftarrow, \bot_\leftarrow\}$. It has two subcases.

Subcase 2.1: $num > num' \geq num''$. By Proposition 2, before the cycle, there has been a 2-chain $\big((x_3, y_3, dir_3, num_3), (x_4, y_4, dir_4, num_4)\big) \in (\Pi_{pub})^2$ such that $y = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4)$. Therefore, $\mathcal{T}^{E,\mathbf{P}}$ creating $(x', y', dir', num')$ gives rise to two 2-chains $\big((x', y'), (x'', y'')\big)$ and $\big((x_3, y_3), (x_4, y_4)\big)$ with $y = x' \oplus \varphi^{-1}(y' \oplus x'') = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4)$. Since $(x', y', dir', num')$ and $(x'', y'', dir'', num'')$ are newly created in this cycle, it holds $num' \geq num'', num_3, num_4$. Therefore, $\mathcal{T}^{E,\mathbf{P}}$ creating $(x', y', dir', num')$ would contradict **Inv4**.

Subcase 2.2: $num > num'' > num'$. For clarity, we list the crucial events in this simulator cycle (in chronological order):

1. $D$ querying $P(x^*)$ or $P(y^*)$ for some $x^*$ or $y^*$ that starts this cycle;

2. $\mathcal{T}^{E,\mathbf{P}}$ creates the record $(x', y', dir', num')$;
3. $\mathcal{T}^{E,\mathbf{P}}$ creates the record $(x'', y'', dir'', num'')$;
4. $\mathcal{T}^{E,\mathbf{P}}$ creates the record $(x, y, dir, num)$.

It could be $(x^*, y^*) = (x', y')$, but $x'' \neq x'$ must hold. Recall that $dir \in \{\leftarrow, \perp_\leftarrow\}$. Thus, by Proposition 2, when $\mathcal{T}^{E,\mathbf{P}}$ is to create $(x'', y'', dir'', num'')$, there has been a 2-chain $\big((x_3, y_3), (x_4, y_4)\big) \in (\Pi_{pub})^2$ such that $y = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4)$. By this and by **Inv4**, it has to be $dir'' \in \{\rightarrow, \perp_\rightarrow\}$. This in turn implies a 2-chain $\big((x_5, y_5), (x_6, y_6)\big) \in (\Pi_{pub})^2$ such that $x'' = y_6 \oplus \varphi(y_5 \oplus x_6)$ by Proposition 2. All the (public) records $(x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_6)$ are created no later than $(x', y')$. Thus, if $(x', y', dir', num')$ is added to $\Pi_{all}$, then the equality $y' \oplus x'' = \varphi(y \oplus x')$ would have caused $\mathcal{T}^{E,\mathbf{P}}$ abort at line 124 in the call to $CheckRecord(x', y', dir', num')$ (and won't add $(x', y')$ to $\Pi_{all}$).

**Case 3: $(x, y, dir, num)$ and $(x', y', dir', num')$ are created in the same cycle occurring after $(x'', y'', dir'', num'')$.** Recall that $dir \in \{\leftarrow, \perp_\leftarrow\}$. The case then resembles **Subcase 2.1**: before $\mathcal{T}^{E,\mathbf{P}}$ creates $(x', y', dir', num')$, (i) the record $(x'', y'', dir'', num'')$ has been in $\Pi_{all}$ (by our assumption), and (ii) by Proposition 2, there has been a 2-chain $\big((x_3, y_3), (x_4, y_4)\big) \in (\Pi_{pub})^2$ such that $y = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4)$. Therefore, $\mathcal{T}^{E,\mathbf{P}}$ creating $(x', y', dir', num')$ with $x' \oplus \varphi^{-1}(y' \oplus x'') = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4)$ contradicts **Inv4**.

**Case 4: $(x, y, dir, num)$ and $(x'', y'', dir'', num'')$ are created in the same cycle occurring after $(x', y', dir', num')$.** This case resembles **Subcase 2.2**. By Proposition 2, when $\mathcal{T}^{E,\mathbf{P}}$ is to create $(x'', y'', dir'', num'')$, there has been a 2-chain $\big((x_3, y_3), (x_4, y_4)\big) \in (\Pi_{pub})^2$ such that $y = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4)$. By this and by **Inv4**, it has to be $dir'' \in \{\rightarrow, \perp_\rightarrow\}$. This in turn implies a 2-chain $\big((x_5, y_5), (x_6, y_6)\big) \in (\Pi_{pub})^2$ with $x'' = y_6 \oplus \varphi(y_5 \oplus x_6)$.

By these, when $CheckInterV3Chain(x', y')$ is called, $(x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_6)$ have all been in $\Pi_{pub}$. Therefore, if $(x', y', dir', num')$ is added to $\Pi_{all}$, then the equality $y' \oplus x'' = \varphi(y \oplus x')$ would have caused $\mathcal{T}^{E,\mathbf{P}}$ abort at line 64 in $CheckInterV3Chain(x', y')$.

**Case 5: $(x, y)$ is created in a separate simulator cycle.** Namely, before the triggering cycle, both $(x', y')$ and $(x'', y'')$ have been in the sets. By Propositions 2 and 1, this means:

(i) At some time in the triggering cycle, there exists another 2-chain $\big((x_2, y_2), (x_3, y_3)\big) \in (\Pi_{pub})^2$ such that $y = x' \oplus \varphi^{-1}(y' \oplus x'') = x_2 \oplus \varphi^{-1}(y_2 \oplus x_3)$; and

(ii) The triggering cycle is due to $D$ querying $P(x_2), P^{-1}(y_2), P(x_3)$ or $P^{-1}(y_3)$.

Note that condition (i) means $(x_2, y_2) \neq (x', y')$ and $(x_3, y_3) \neq (x'', y'')$: the former is obvious, while the latter is ensured by **Inv1**. In addition, since $(x', y')$ remains internal in this cycle, it also holds $(x', y') \neq (x_3, y_3)$. For the remaining, we address two subcases:

*Subcase 5.1: the triggering cycle is a new cycle.* This is impossible, since: if the triggering cycle is due to $D$ querying $P(x_2), P^{-1}(y_2)$ or $P^{-1}(y_3)$, then $x' \oplus \varphi^{-1}(y' \oplus x'') = x_2 \oplus \varphi^{-1}(y_2 \oplus x_3)$ contradicts **Inv4**; if the triggering cycle is due to $D$ querying $P(x_3)$, then $\mathcal{T}^{E,\mathbf{P}}$ would have aborted at line 41 in $CheckPrivacy(x_3)$.

*Subcase 5.2: the triggering cycle is a transferring cycle.* This is impossible either:

- If the triggering cycle is due to $D$ querying $P(x_2)$, then $dir_2 \in \{\rightarrow, \perp_\rightarrow\}$, as otherwise $\mathcal{T}^{E,\mathbf{P}}$ would have aborted at line 37 in $CheckPrivacy(x_2)$. Then, the equality $x' \oplus \varphi^{-1}(y' \oplus x'') = x_2 \oplus \varphi^{-1}(y_2 \oplus x_3)$ would have caused $\mathcal{T}^{E,\mathbf{P}}$ abort at line 52 in $CheckInternalColl(x_2, y_2)$ (and $\mathcal{T}^{E,\mathbf{P}}$ won't create the purported record $(x, y)$);
- If the triggering cycle is due to $D$ querying $P^{-1}(y_2)$, then $dir_2 \in \{\leftarrow, \perp_\leftarrow\}$, as otherwise $\mathcal{T}^{E,\mathbf{P}}$ aborted at line 43 in $CheckPrivacy^{-1}(y_2)$. Then, $x' \oplus \varphi^{-1}(y' \oplus x'') = x_2 \oplus \varphi^{-1}(y_2 \oplus x_3)$ would cause $\mathcal{T}^{E,\mathbf{P}}$ abort at line 61 in $CheckInternalColl^{-1}(x_2, y_2)$;

- If the triggering cycle is due to $D$ querying $P(x_3)$, then $\mathcal{T}^{E,\mathbf{P}}$ would have aborted at line 41 in $CheckPrivacy(x_3)$ (and won't create $(x, y)$);
- Finally, if the triggering cycle is due to $D$ querying $P^{-1}(y_3)$, then $dir_3 \in \{\leftarrow, \perp_{\leftarrow}\}$—and $x' \oplus \varphi^{-1}(y' \oplus x'') = x_2 \oplus \varphi^{-1}(y_2 \oplus x_3)$ would have caused $\mathcal{T}^{E,\mathbf{P}}$ abort at line 59 in the call to $CheckInternalColl^{-1}(x_3, y_3)$.

The above have excluded all possibilities of creating bad 3-chains. Thus the claim.

□

## 5 Simulator Complexity

In this section, we establish upper bounds on the complexity of the simulator.

**Lemma 3.** *In any $\Sigma_2$ execution, after the q-th simulator cycle returns, it holds*

1. *The number of calls to the procedures $Complete^+$ and $Complete^-$ that have been made by $\mathcal{T}^{E,\mathbf{P}}$ is at most $q^2$ in total;*
2. *$|\Pi_{pub}| \le q$, $|\Pi_{in}| \le 2q^2$, $|ET| \le q^2$, and $\mathcal{T}^{E,\mathbf{P}}$ runs in time $O(q^2)$.*

*In addition, these bounds hold for the simulator $\mathcal{S}$ in any $\Sigma_1$ execution.*

*Proof.* First, $|\Pi_{pub}|$ increases by at most 1 upon each adversarial query (to $P(x)$ or $P^{-1}(y)$), and stays invariant otherwise. Thus, $|\Pi_{pub}| \le q$ after the $q$-th cycle.

For the remaining, we analyze the simulator cycles in detail. First, consider the case where the $j$-th adversarial query is forward $P(x^{(j)})$. It holds $|\Pi_{pub}| \le j-1$ before the cycle. Regardless of $x^{(j)} \in domain(\Pi_{all})$ or not before the cycle, $\mathcal{T}^{E,\mathbf{P}}$ adds a record $(x^{(j)}, y^{(j)}, \rightarrow, n^{(j)})$ to $\Pi_{pub}$. By the design of $Check3Chains$ and $Check2Chains$ (see Sect. 3), every subsequently detected 2-chain $\big((x^{(j)}, y^{(j)}), (x', y')\big)$ is associated with a unique $(x', y') \in \Pi_{pub}$, $(x', y') \ne (x^{(j)}, y^{(j)})$. Thus, $\mathcal{T}^{E,\mathbf{P}}$ detects $\le |\Pi_{pub}| \le j-1$ such 2-chains. Every subsequently detected 2-chain $\big((x', y'), (x^{(j)}, y^{(j)})\big)$ or 3-chain $\big((x, y), (x', y'), (x^{(j)}, y^{(j)})\big)$ with $(x', y') \in \Pi_{pub}$ is associated with a unique with $(x', y') \in \Pi_{pub}$. Thus, $\mathcal{T}^{E,\mathbf{P}}$ detects $\le j$ such 2-chains and 3-chains. For each detected 3-chain/2-chain, $\mathcal{T}^{E,\mathbf{P}}$ makes 1 call to $Complete^+/Complete^-$, at most 1 call to $InP/InP^{-1}$, and at most 1 call to $Adapt$. Therefore, the $j$-th simulator cycle due to $P(x^{(j)})$ makes at most $2j-1$ calls to $Complete^+/Complete^-$, adds at most $2j-1$ internal records to $\Pi_{in}$, makes at most $2j-1$ call to $Adapt$ (thus adding at most $2j-1$ adapted records to $\Pi_{in}$) and at most $2j-1$ queries to $E$ (thus adding at most $2j-1$ records to $ET$).

When the $j$-th adversarial query is backward $P^{-1}(y^{(j)})$, the analysis is similar by symmetry and yields the same bound. Summing over the $q$ queries yields

$$|\Pi_{in}| \le \sum_{j=1}^{q} (2j-1) + \sum_{j=1}^{q} (2j-1) \le 2q^2, |ET| \le \sum_{j=1}^{q} 2j-1 \le q^2. \tag{2}$$

The running time is dominated by $Complete^+/Complete^-$, and is $O(q^2)$. Finally, since $\mathcal{S}^{E,\mathbf{P}}$ has no more actions than $\mathcal{T}^{E,\mathbf{P}}$, the bounds hold for $\mathcal{S}^{E,\mathbf{P}}$ in any $\Sigma_1$ execution.

□

## 6 Treatments for Internal Records

As mentioned in the Introduction, a central intuition is that every internal record in $\Pi_{in}$ has one "endpoint" that is kept "private" to the distinguisher. We now provide a formal treatment. The underlying idea resembles that of [DRST12], i.e., certain "internal" randomness has no influence on the actions of the simulator.

**Lemma 4.** *Assume that* $\Pr_{E,\mathbf{p}}\big[D^{EMSP[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}},\mathcal{T}^{E,\mathbf{p}}}\ aborts\big] \leq 1/2$. *Let* $\Pi_{all}^{(\ell)}, \Pi_{pub}^{(\ell)}$ *and* $\Pi_{in}^{(\ell)}$ *be the sets of* $\mathcal{T}^{E,\mathbf{p}}$ *that stand after* $\mathcal{T}^{E,\mathbf{p}}$ *completes its* $\ell$*-th simulator cycle in the execution* $D^{EMSP[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}},\mathcal{T}^{E,\mathbf{p}}}$. *Then, at the end of the* $\ell$*-th simulator cycle, consider a record* $(x^\circ, y^\circ, dir^\circ, num^\circ) \in \Pi_{in}^{(\ell)}$. *As long as* $\mathcal{T}^{E,\mathbf{p}}$ *has not aborted, it holds:*

- *If* $dir^\circ = \rightarrow$ *or* $\bot_\rightarrow$, *then conditioned on the transcript of queries and responses already obtained by* $D$, $x^\circ$ *is fixed; conditioned on the* $\big|\Pi_{all}^{(\ell)}\big| - 1$ *records in* $\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}$, $y^\circ$ *remains uniformly distributed in a set of size at least* $2^n/2 - 3q^2$;
- *If* $dir^\circ = \leftarrow$ *or* $\bot_\leftarrow$, *then conditioned on the transcript of queries and responses already obtained by* $D$, $y^\circ$ *is fixed; conditioned on the* $\big|\Pi_{all}^{(\ell)}\big| - 1$ *records in* $\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}$, $x^\circ$ *remains uniformly distributed in a set of size at least* $2^n/2 - 3q^2$.

*Consequently,*

- *the probability that* $\mathcal{T}^{E,\mathbf{p}}$ *aborts in the procedures CheckPrivacy and CheckPrivacy*$^{-1}$ *is at most* $\frac{8q^3}{2^n} + \frac{36C(\varphi)q^5}{2^n} + \frac{108C(\varphi)q^7}{2^n}$ *in total;*
- *the probability that* $\mathcal{T}^{E,\mathbf{p}}$ *aborts in CheckInternalColl and CheckInternalColl*$^{-1}$ *is at most* $\frac{36C(\varphi)q^5}{2^n} + \frac{324C(\varphi)q^7}{2^n}$ *in total;*
- *the probability that* $\mathcal{T}^{E,\mathbf{p}}$ *aborts in CheckInterV3Chain is at most* $\frac{12q^7}{2^n}$ *in total.*

**Proof Setup.** We view the combination of $D$ and $\mathrm{EMSP}[\varphi]_4$ as a single adversary $B$ that interacts with $\mathcal{T}^{E,\mathbf{p}}$, and write $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}}$ instead of $D^{\mathrm{EMSP}[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}},\mathcal{T}^{E,\mathbf{p}}}$. Since $D$ is deterministic (see Sect. 4), $B$ is also deterministic.

Then, consider an arbitrary record $(x^\circ, y^\circ, dir^\circ, num^\circ) \in \Pi_{in}^{(\ell)}$. We proceed by showing that we can replace the "private" endpoint $y^\circ$ (when $dir^\circ \in \{\rightarrow, \bot_\rightarrow\}$) or $x^\circ$ (when $dir^\circ \in \{\leftarrow, \bot_\leftarrow\}$) without affecting the "main actions" in the $\Sigma_2$ execution *and* the transcript of $B$. To formalize this idea, we consider a modified $\Sigma_2$ execution $D^{\Sigma_2(\mathrm{EMSP}[\varphi]_4^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}},\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi})}$ (also denoted $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}}$) capturing the interaction between $D$ and $\Sigma_2(\mathrm{EMSP}[\varphi]_4^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}},\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi})$ that uses a random permutation $\pi$ as the randomness and $\mathrm{EMSP}[\varphi]_4^\pi$ instead of the ideal cipher $E$. $\pi$ is *good*, if:

- $\pi \vdash \big(\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}\big)$, i.e., $\pi(x) = y$ if and only if $(x, y) \in \big(\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}\big)$; and
- The $\Sigma_2$ execution $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}}$ does not abort.

At any time during the executions, we say that the set $\Pi_{all} = \Pi_{pub} \cup \Pi_{in}$ of $\mathcal{T}^{E,\mathbf{p}}$ and the set $\Pi'_{all}$ of $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ are *isomorphic w.r.t.* $(x^\circ, y^\circ, dir^\circ)$, denoted $\Pi_{all} \cong \Pi'_{all}$, if:

- When $dir^\circ \in \{\bot_\rightarrow, \rightarrow\}$ and $(x^\circ, y^\circ, dir^\circ) \in \Pi_{in}$, it holds $(x^\circ, y^{\circ\circ}, dir^\circ) \in \Pi'_{in}$ correspondingly, where $y^{\circ\circ} = \pi(x^\circ)$.

  Otherwise, $\Pi_{pub} = \Pi'_{pub}$, $\big(\Pi_{in} \backslash \{(x^\circ, y^\circ)\}\big) = \big(\Pi'_{in} \backslash \{(x^\circ, y^{\circ\circ})\}\big)$;
- When $dir^\circ \in \{\leftarrow, \bot_\leftarrow\}$ and $(x^\circ, y^\circ, dir^\circ) \in \Pi_{in}$, it holds $(x^{\circ\circ}, y^\circ, dir^\circ) \in \Pi'_{in}$ correspondingly, where $x^{\circ\circ} = \pi^{-1}(y^\circ)$;

  Otherwise, $\Pi_{pub} = \Pi'_{pub}$, $\big(\Pi_{in} \backslash \{(x^\circ, y^\circ)\}\big) = \big(\Pi'_{in} \backslash \{(x^{\circ\circ}, y^\circ)\}\big)$.

With the above, in Sect. 6.1 we show that, using any good permutation $\pi$, the modified $\Sigma_2$ execution $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}}$ and the original $B^{\mathcal{T}^{E,\mathbf{p}}}$ are "isomorphic", meaning that after the $j$-th cycle, $j = 1, ..., \ell$, the sets of $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ and $\mathcal{T}^{E,\mathbf{p}}$ are always isomorphic w.r.t. $(x^\circ, y^\circ, dir^\circ)$, i.e., $\Pi_{all}^{(j)} \cong (\Pi'_{all})^{(j)}$. This particularly means $\Pi_{pub}^{(j)} \cong (\Pi'_{pub})^{(j)}$ always holds, and $B$ has the same transcript of queries and responses—and further, $B$ cannot decide which execution it is in. We then argue in Sect. 6.1.4 that, conditioned on that $\pi$ is good, either $x^\circ$ or $y^\circ$ in the record $(x^\circ, y^\circ, dir^\circ) \in \Pi_{in}$ is uniformly distributed in at least $2^n/2 - 3q^2$ possibilities. This enables bounding the abort probabilities of *CheckPrivacy*, *CheckInternalColl* and *CheckInterV3Chain* in Sect. 6.2, 6.3 and 6.4.

## 6.1 For any good $\pi$, $B^{\mathcal{T}^{\mathsf{EMSP}[\varphi]_4^{\pi},\pi}}$ and $B^{\mathcal{T}^{E,\mathbf{p}}}$ are "isomorphic"

Imagine executing $B^{\mathcal{T}^{E,\mathbf{p}}}$ and $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}}$ in parallel. It is easy to see that the query records (even if adapted) created by $\mathcal{T}^{E,\mathbf{p}}$ in $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}}$ are *always* consistent with $\pi$. But this does not necessarily mean $\mathcal{T}^{E,\mathbf{p}}$ and $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}$ will create almost-identical sets. To prove, we use an induction over simulator cycles.

With the above in mind, for any $j \in \{1, ..., \ell\}$, consider the point right before $B = D^{\mathrm{EMSP}[\varphi]_4}$ issues its $j$-th query to start the $j$-th simulator cycle. Wlog, assume that the query is forward $P(x^{(j)})$. Further assuming $\Pi_{all}^{(j-1)} \cong (\Pi'_{all})^{(j-1)}$, i.e., $\Pi_{all}^{(j-1)}$ and $(\Pi'_{all})^{(j-1)}$ are "isomorphic".

### 6.1.1 Case 1: $(x^\circ, y^\circ) \notin \Pi_{all}^{(j-1)}$, and $(x^\circ, y^\circ) \notin \Pi_{all}^{(j)}$

By definition, $\Pi_{all}^{(j-1)} \cong (\Pi'_{all})^{(j-1)}$ implies $\Pi_{pub}^{(j-1)} = (\Pi'_{pub})^{(j-1)}$ and $\Pi_{in}^{(j-1)} = (\Pi'_{in})^{(j-1)}$. In this case, the analysis is totally free of the influences of $(x^\circ, y^\circ)$. To see this, we consider the concrete actions in this simulator cycle in turn.

**Initial step.** The concrete initial action distinguishes two cases.

- **Case 1.1.a:** $x^{(j)} \notin domain(\Pi_{all}^{(j-1)})$. Then $\mathcal{T}^{E,\mathbf{p}}$ "downloads" $y^{(j)} \leftarrow \mathbf{p}(x^{(j)})$ and creates $(x^{(j)}, y^{(j)}, \rightarrow)$. By this, $\Pi_{pub}^{(j)} = \Pi_{pub}^{(j-1)} \cup \{(x^{(j)}, y^{(j)}, \rightarrow)\}$.

  Since $\Pi_{all}^{(j-1)} \cong (\Pi'_{all})^{(j-1)}$, it also holds $x^{(j)} \notin domain\big((\Pi'_{all})^{(j-1)}\big)$, and $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}$ "downloads" $(y')^{(j)} \leftarrow \pi(x^{(j)})$. Since $(x^{(j)}, y^{(j)}) \in \Pi_{pub}^{(j)} \subseteq \Pi_{pub}^{(\ell)}$ and since $\pi \vdash \Pi_{pub}^{(\ell)}$, it holds $(y')^{(j)} = y^{(j)}$, and further $(\Pi'_{pub})^{(j)} = (\Pi'_{pub})^{(j-1)} \cup \{(x^{(j)}, y^{(j)}, \rightarrow)\}$.

- **Case 1.1.b:** $(x^{(j)}, y^{(j)}, dir^{(j)}) \in \Pi_{in}^{(j-1)}$. Then $\mathcal{T}^{E,\mathbf{p}}$ moves the record $(x^{(j)}, y^{(j)}, dir^{(j)})$ to $\Pi_{pub}$. Since $\Pi_{in}^{(j-1)} = (\Pi'_{in})^{(j-1)}$, it also holds $(x^{(j)}, y^{(j)}, dir^{(j)}) \in (\Pi'_{in})^{(j-1)}$. Thus, $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}$ also moves $(x^{(j)}, y^{(j)}, dir^{(j)})$ to $\Pi'_{pub}$.

The above means $(\Pi'_{pub})^{(j)} = \Pi_{pub}^{(j)}$ still holds after the $j$-th simulator cycle.

For convenience, let $\Pi_{tmp}^{(j-1)} := \Pi_{pub}^{(j)} \cup \Pi_{in}^{(j-1)}$ and $(\Pi'_{tmp})^{(j-1)} := (\Pi'_{pub})^{(j)} \cup (\Pi'_{in})^{(j-1)}$. It is easy to see $\Pi_{tmp}^{(j-1)} = (\Pi'_{tmp})^{(j-1)}$ and $(x^\circ, y^\circ) \notin \Pi_{tmp}^{(j-1)}$.

**Chain detection and completion.** Two types of structures are relevant in this phase.

*For every 2-chain $\big((x_2, y_2), (x^{(j)}, y^{(j)})\big)$, $(x_2, y_2) \in \Pi_{pub}^{(j)}$:* Let $y_1 = x_2 \oplus \varphi^{-1}(y_2 \oplus x^{(j)})$. Since $(\Pi'_{pub})^{(j)} = \Pi_{pub}^{(j)}$ (as we just showed), it also holds $(x_2, y_2) \in (\Pi'_{pub})^{(j)}$. For the remaining actions, we further distinguish two cases as follows.

- **Case 1.2.a:** $(x_1, y_1) \in \Pi_{tmp}^{(j)}$. Since $(\Pi'_{tmp})^{(j)} = \Pi_{tmp}^{(j)}$, it also holds $(x_1, y_1) \in (\Pi'_{tmp})^{(j)}$. The simulator $\mathcal{T}^{E,\mathbf{p}}$ in $B^{\mathcal{T}^{E,\mathbf{p}}}$ thus detects a 3-chain, and sets $k \leftarrow \varphi^{-1}(y_1 \oplus x_2)$, $u \leftarrow x_1 \oplus k$, $x_4 \leftarrow y^{(j)} \oplus \varphi^3(k)$, queries $E(k, u) \rightarrow v$, sets $y_4 \leftarrow v \oplus \varphi^4(k)$ and adds an adapted record $(x_4, y_4, \bot_{\rightarrow})$ to $\Pi_{in}$ to complete the 4-chain

$$\big((x_1, y_1), (x_2, y_2), (x^{(j)}, y^{(j)}), (x_4, y_4, \bot_{\rightarrow})\big).$$

Clearly, $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ in $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}}$ also detects $\big((x_1, y_1), (x_2, y_2), (x^{(j)}, y^{(j)})\big)$: in the view of $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$, $(x^{(j)}, y^{(j)})$ is also newly added to $\Pi'_{pub}$, and $(x_1, y_1) \in (\Pi'_{tmp})^{(j)}$ and $(x_2, y_2) \in (\Pi'_{pub})^{(j)}$ also hold. To handle this 3-chain, $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ computes the same $k$ and $u$, queries $\mathrm{EMSP}[\varphi]_4^{\mathbf{P}}(k, u) \to v'$, sets $y'_4 \leftarrow v' \oplus \varphi^4(k)$ and adds $(x_4, y'_4, \bot_\to)$ to $\Pi'_{in}$ to complete the 4-chain (if abortion never occurs)

$$\big((x_1, y_1), (x_2, y_2), (x^{(j)}, y^{(j)}), (x_4, y'_4, \bot_\to)\big).$$

As mentioned, it holds $y'_4 = \pi(x_4)$. Since $(x^\circ, y^\circ) \notin \Pi_{tmp}^{(j-1)}$, it must be $x_4 \neq x^\circ$. By this and by $\pi \vdash \big(\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}\big)$, it holds $y'_4 = y_4$, and $\Pi'_{in} = \Pi_{in}$. Therefore, $\Pi_{all} \cong \Pi'_{all}$ still holds after completing this 4-chain.

- **Case 1.2.b:** $y_1 \notin range(\Pi_{tmp}^{(j-1)})$. In this case, $\mathcal{T}^{E,\mathbf{P}}$ detects a 2-chain in the 1st execution $D^{\mathrm{EMSP}[\varphi]_4^{\mathcal{T}^{E,\mathbf{P}}},\mathcal{T}^{E,\mathbf{P}}}$ and completes a 4-chain

  $$\big((x_1, y_1, \bot_\leftarrow), (x_2, y_2), (x^{(j)}, y^{(j)}), (x_4, y_4, \to)\big).$$

  Thus, $\mathcal{T}^{E,\mathbf{P}}$ adds $(x_4, y_4, \to)$ and $(x_1, y_1, \bot_\leftarrow)$ to $\Pi_{in}$.

  Since $(\Pi'_{tmp})^{(j-1)} = \Pi_{tmp}^{(j-1)}$, it holds $y_1 \notin range\big((\Pi'_{tmp})^{(j-1)}\big)$ as well, and $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ also detects a 2-chain and complete a 4-chain

  $$\big((x'_1, y_1, \bot_\leftarrow), (x_2, y_2), (x^{(j)}, y^{(j)}), (x_4, y'_4, \to)\big)$$

  with $x'_1 = \pi^{-1}(y_1)$ and $y'_4 = \pi(x_4)$. Since $(x^\circ, y^\circ) \notin \Pi_{tmp}^{(j-1)}$, it must be $x_1 \neq x^\circ$ and $x_4 \neq x^\circ$. By this and by $\pi \vdash \big(\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}\big)$, it holds $x'_1 = x_1$, $y'_4 = y_4$; $\Pi'_{in} = \Pi_{in}$ and thus $\Pi_{all} \cong \Pi'_{all}$ after completing this 4-chain.

*For every 2-chain $\big((x^{(j)}, y^{(j)}), (x_3, y_3)\big)$, $(x_3, y_3) \in \Pi_{pub}^{(j)}$:* The argument is similar to Case 1.2.b above. Namely, $\mathcal{T}^{E,\mathbf{P}}$ detects a 2-chain $\big((x^{(j)}, y^{(j)}), (x_3, y_3)\big)$ and adds $(x_1, y_1, \leftarrow)$ and $(x_4, y_4, \bot_\to)$ to $\Pi_{in}$ to complete a 4-chain

$$\big((x_1, y_1, \leftarrow), (x^{(j)}, y^{(j)}), (x_3, y_3), (x_4, y_4, \bot_\to)\big),$$

while $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ also detects $\big((x^{(j)}, y^{(j)}), (x_3, y_3)\big)$ and complete a 4-chain

$$\big((x'_1, y_1, \leftarrow), (x^{(j)}, y^{(j)}), (x_3, y_3), (x_4, y'_4, \bot_\to)\big)$$

with $x'_1 = \pi^{-1}(y_1) = x_1$ and $y'_4 = \pi(x_4) = y_4$. Therefore, after completing this 4-chain, $\Pi'_{in} = \Pi_{in}$ and $\Pi_{all} \cong \Pi'_{all}$ still holds.

**In summary,** if: (i) $(\Pi'_{all})^{(j-1)} \cong \Pi_{all}^{(j-1)}$, and (ii) $(x^\circ, y^\circ) \notin \Pi_{all}^{(j-1)}$ and $(x^\circ, y^\circ) \notin \Pi_{all}^{(j)}$, then in the subsequent $j$-th simulator cycle,

- every time $\mathcal{T}^{E,\mathbf{P}}$ adds a record $(x^{(j)}, y^{(j)})$ to its set $\Pi_{pub}$, $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ adds the same record $(x^{(j)}, y^{(j)})$ to its set $\Pi'_{pub}$;

- every time $\mathcal{T}^{E,\mathbf{P}}$ adds a record $(x, y, dir)$ to its set $\Pi_{in}$, $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ adds the same record $(x, y, dir)$ to its set $\Pi'_{in}$.

The above mean $(\Pi'_{all})^{(j)} \cong \Pi_{all}^{(j)}$ after the $j$-th cycle.

### 6.1.2 Case 2: $(x^\circ, y^\circ) \notin \Pi_{all}^{(j-1)}$, and $(x^\circ, y^\circ) \in \Pi_{all}^{(j)}$

By the definition of "isomorphic", this means $\Pi_{pub}^{(j-1)} = (\Pi'_{pub})^{(j-1)}$, $\Pi_{in}^{(j-1)} = (\Pi'_{in})^{(j-1)}$. This case contains the "interesting" action of creating the record $(x^\circ, y^\circ, dir^\circ)$.

Below we consider the concrete actions in this simulator cycle in turn.

**Initial step.** Similarly to Case 1, since $\Pi_{pub}^{(j-1)} = (\Pi'_{pub})^{(j-1)}$ and $\Pi_{in}^{(j-1)} = (\Pi'_{in})^{(j-1)}$, $\mathcal{T}^{E,\mathbf{P}}$ and $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ have the same initial step, and $(\Pi'_{pub})^{(j)} = \Pi_{pub}^{(j)}$ still holds. Let $\Pi_{tmp}^{(j-1)} := \Pi_{pub}^{(j)} \cup \Pi_{in}^{(j-1)}$ and $(\Pi'_{tmp})^{(j-1)} := (\Pi'_{pub})^{(j)} \cup (\Pi'_{in})^{(j-1)}$ for the remaining argument in Case 2. Still, $\Pi_{tmp}^{(j-1)} = (\Pi'_{tmp})^{(j-1)}$.

**Chain detection and completion.** Two types of structures are relevant in this phase. *For every 2-chain $\big((x_2, y_2), (x^{(j)}, y^{(j)})\big)$, $(x_2, y_2) \in \Pi_{pub}^{(j)}$:* Let $y_1 = x_2 \oplus \varphi^{-1}(y_2 \oplus x^{(j)})$. Since $(\Pi'_{pub})^{(j)} = \Pi_{pub}^{(j)}$ (as we just showed), it also holds $(x_2, y_2) \in (\Pi'_{pub})^{(j)}$. For the remaining actions, we further distinguish two cases as follows.

- **Case 2.2.a:** $(x_1, y_1) \in \Pi_{tmp}^{(j-1)}$. This case is similar to Case 1.2.a: the simulator $\mathcal{T}^{E,\mathbf{P}}$ in $B^{\mathcal{T}^{E,\mathbf{P}}}$ detects a 3-chain and adds an adapted record $(x_4, y_4, \bot_\rightarrow)$ to $\Pi_{in}$ to complete the 4-chain $\big((x_1, y_1), (x_2, y_2), (x^{(j)}, y^{(j)}), (x_4, y_4, \bot_\rightarrow)\big)$. Meanwhile, $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ in $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}}$ also detects $\big((x_1, y_1), (x_2, y_2), (x^{(j)}, y^{(j)})\big)$ and adds $(x_4, y'_4, \bot_\rightarrow)$ to $\Pi'_{in}$ to complete 4-chain $\big((x_1, y_1), (x_2, y_2), (x^{(j)}, y^{(j)}), (x_4, y'_4, \bot_\rightarrow)\big)$ (if abortion never occurs) with $y'_4 = \pi(x_4)$. At this stage,
  - If $dir^\circ \in \{\leftarrow, \bot_\leftarrow\}$, then the creation of $(x_4, y_4, \bot_\rightarrow)$ is irrelevant to our focus $(x^\circ, y^\circ)$. In a similar vein to Case 1.2.a, we simply have $y'_4 = y_4$ and the sets $\Pi_{in}$ and $\Pi'_{in}$ still have the same contents after the creation of $(x_4, y'_4, \bot_\rightarrow)$;
  - If $dir^\circ \in \{\rightarrow, \bot_\rightarrow\}$ though $x_4 \neq x^\circ$, then by this and by $\pi \vdash \big(\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}\big)$, it holds $y'_4 = y_4$ and $\Pi_{in} = \Pi'_{in}$ after creating $(x_4, y'_4, \bot_\rightarrow)$;
  - Else, i.e., $x_4 = x^\circ$, then $\mathcal{T}^{E,\mathbf{P}}$ adds $(x^\circ, y^{\circ\circ}, \bot_\rightarrow)$ to $\Pi'_{in}$, while $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ adds $(x^\circ, y^{\circ\circ}, \bot_\rightarrow)$, $y^{\circ\circ} = \pi(x^\circ)$, to $\Pi'_{in}$.

  Therefore, it remains $\Pi'_{in} \cong \Pi_{in}$ after completing this 4-chain.

- **Case 2.2.b:** $y_1 \notin range(\Pi_{all}^{(j)})$. This case is similar to Case 1.2.b: $\mathcal{T}^{E,\mathbf{P}}$ detects a 2-chain $\big((x_2, y_2), (x^{(j)}, y^{(j)})\big)$ in $B^{\mathcal{T}^{E,\mathbf{P}}}$ and completes a 4-chain $\big((x_1, y_1, \bot_\leftarrow), (x_2, y_2), (x^{(j)}, y^{(j)}), (x_4, y_4, \rightarrow)\big)$. Since $(\Pi'_{tmp})^{(j-1)} = \Pi_{tmp}^{(j-1)}$, it holds $y_1 \notin range\big((\Pi'_{tmp})^{(j-1)}\big)$ as well, and $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ also detects a 2-chain and complete a 4-chain $\big((x'_1, y_1, \bot_\leftarrow), (x_2, y_2), (x^{(j)}, y^{(j)}), (x_4, y'_4, \rightarrow)\big)$ with $x'_1 = \pi^{-1}(y_1)$ and $y'_4 = \pi(x_4)$. Now,
  - If $dir^\circ \in \{\leftarrow, \bot_\leftarrow\}$, then $(x_4, y_4, \rightarrow)$ is irrelevant with $(x^\circ, y^\circ)$. Whereas,
    - ∗ If $y_1 = y^\circ$, then $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ adds $(x^{\circ\circ}, y^\circ, \bot_\leftarrow)$, $x^{\circ\circ} = \pi^{-1}(y^\circ)$, to $\Pi'_{in}$;
    - ∗ Else, i.e., $y_1 \neq y^\circ$, then by $\pi \vdash \big(\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}\big)$ we have $x'_1 = x_1$ and $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ adds (the same record as $\mathcal{T}^{E,\mathbf{P}}$) $(x_1, y_1, \bot_\leftarrow)$ to $\Pi'_{in}$.

    Therefore, it holds $\Pi_{in} \cong \Pi'_{in}$ after completing this chain.
  - If $dir^\circ \in \{\rightarrow, \bot_\rightarrow\}$, then $(x_1, y_1, \bot_\leftarrow)$ is irrelevant with $(x^\circ, y^\circ)$. Whereas,
    - ∗ If $x_4 = x^\circ$, then $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ adds $(x^\circ, y^{\circ\circ}, \rightarrow)$, $y^{\circ\circ} = \pi(x^\circ)$, to $\Pi'_{in}$;
    - ∗ Else, i.e., $x_4 \neq x^\circ$, then by $\pi \vdash \big(\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}\big)$ we have $y'_4 = y_4$ and $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^\pi,\pi}$ adds (the same record as $\mathcal{T}^{E,\mathbf{P}}$) $(x_4, y_4, \rightarrow)$ to $\Pi'_{in}$.

Therefore, it holds $\Pi_{in} \cong \Pi'_{in}$ after completing this chain.

By the above, in any (sub)case, $\Pi_{all} \cong \Pi'_{all}$ still holds after completing this 4-chain.

*For every 2-chain $\big((x^{(j)}, y^{(j)}), (x_3, y_3)\big)$, $(x_3, y_3) \in \Pi_{pub}^{(j)}$:* The argument is similar to Case 2.2.b above. $\mathcal{T}^{E,\mathbf{P}}$ detects a 2-chain $\big((x^{(j)}, y^{(j)}), (x_3, y_3)\big)$ and complete a 4-chain $\big((x_1, y_1, \leftarrow), (x^{(j)}, y^{(j)}), (x_3, y_3), (x_4, y_4, \bot_{\rightarrow})\big)$. Whereas $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}$ also detects a 2-chain and completes a 4-chain $\big((x'_1, y_1, \leftarrow), (x^{(j)}, y^{(j)}), (x_3, y_3), (x_4, y'_4, \bot_{\rightarrow})\big)$ with $x'_1 = \pi^{-1}(y_1) = x_1$ and $y'_4 = \pi(x_4) = y_4$. In a similar vein to Case 2.2.b, it can be shown $\Pi_{all} \cong \Pi'_{all}$ still holds after completing this 4-chain.

**In summary,** if: (i) $(\Pi'_{all})^{(j-1)} \cong \Pi_{all}^{(j-1)}$, and (ii) $(x^\circ, y^\circ) \notin \Pi_{all}^{(j-1)}$ while $(x^\circ, y^\circ) \in \Pi_{all}^{(j)}$, then in the subsequent $j$-th simulator cycle,

- every time $\mathcal{T}^{E,\mathbf{P}}$ adds a record $(x^{(j)}, y^{(j)})$ to its set $\Pi_{pub}$, $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}$ adds the same record $(x^{(j)}, y^{(j)})$ to its set $\Pi'_{pub}$;
- For any $(x, y) \neq (x^\circ, y^\circ)$, every time $\mathcal{T}^{E,\mathbf{P}}$ adds a record $(x, y, dir)$ to its set $\Pi_{in}$, $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}$ adds the same record $(x, y, dir)$ to its set $\Pi'_{in}$;
- If $dir^\circ \in \{\rightarrow, \bot_{\rightarrow}\}$ then $(x^\circ, y^{\circ\circ}, dir^\circ) \in (\Pi'_{in})^{(j)}$, where $y^{\circ\circ} = \pi(x^\circ)$; if $dir^\circ \in \{\leftarrow, \bot_{\leftarrow}\}$ then $(x^{\circ\circ}, y^\circ, dir^\circ) \in (\Pi'_{in})^{(j)}$, $x^{\circ\circ} = \pi^{-1}(y^\circ)$.

The above mean $(\Pi'_{all})^{(j)} \cong \Pi_{all}^{(j)}$ after the $j$-th cycle.

### 6.1.3 Case 3: $(x^\circ, y^\circ) \in \Pi_{all}^{(j-1)}$

In this case, the crux is to show that the minor difference between $\Pi_{in}^{(j-1)}$ and $(\Pi'_{in})^{(j-1)}$ will not affect the simulator actions. To this end, the **crucial** property (of our simulators) is that *they never "intentionally" create records on the "private side" of the internal record* $(x^\circ, y^\circ)$ (as reflected by Proposition 2), so that the difference on "private sides" have no essential influence. Below we consider the actions in this simulator cycle in turn.

**Initial step.** The concrete initial action distinguishes two cases.

- **Case 3.1.a:** $x^{(j)} \notin domain(\Pi_{all}^{(j-1)})$. We argue $x^{(j)} \notin domain\big((\Pi'_{all})^{(j-1)}\big)$. Assume otherwise, then the only possibility is:
  - $x^{(j)} = x^\circ$ for the internal record $(x^\circ, y^\circ, dir^\circ) \in \Pi_{in}^{(j-1)}$, and
  - $dir \in \{\leftarrow, \bot_{\leftarrow}\}$, so that $(x^{\circ\circ}, y^\circ, dir^\circ) \in \Pi_{in}^{(j-1)}$ for some $x^{\circ\circ} \neq x^\circ$.

  But then, $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}$ would have aborted at line 37 in the call $CheckPrivacy(x^{(j)})$, and this contradicts our assumption that $\pi$ is good and $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}}$ does not abort. Thus, in $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}}$ it holds $x^{(j)} \notin domain\big((\Pi'_{all})^{(j-1)}\big)$ as well, and both $\mathcal{T}^{E,\mathbf{P}}$ and $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}$ "download" $y^{(j)} \leftarrow \mathbf{p}(x^{(j)})$ or $y^{(j)} \leftarrow \pi(x^{(j)})$ and create $(x^{(j)}, y^{(j)}, \rightarrow)$, so that $(\Pi'_{pub})^{(j)} = \Pi_{pub}^{(j)} = \Pi_{pub}^{(j-1)} \cup \{(x^{(j)}, y^{(j)}, \rightarrow)\}$.

- **Case 3.1.b:** $(x^{(j)}, y^{(j)}, dir^{(j)}) \in \Pi_{in}^{(j-1)}$. It holds $dir^{(j)} \in \{\rightarrow, \bot_{\rightarrow}\}$, otherwise $D$ querying $P(x^{(j)})$ would have caused abort at line 37 in $CheckPrivacy(x^{(j)})$. $\mathcal{T}^{E,\mathbf{P}}$ then moves $(x^{(j)}, y^{(j)}, dir^{(j)})$ to $\Pi_{pub}$, thus $\Pi_{pub}^{(j)} = \Pi_{pub}^{(j-1)} \cup \{(x^{(j)}, y^{(j)}, \rightarrow)\}$.

  Since we assumed $(x^\circ, y^\circ, dir^\circ) \in \Pi_{in}^{(\ell)}$, it holds $(x^{(j)}, y^{(j)}) \neq (x^\circ, y^\circ)$. Since $\Pi_{all}^{(j-1)} \cong (\Pi'_{all})^{(j-1)}$ and $(x^{(j)}, y^{(j)}) \neq (x^\circ, y^\circ)$, it also holds $(x^{(j)}, y^{(j)}, dir^{(j)}) \in (\Pi_{in}^{(j-1)})'$. Therefore, in $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}}$, $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}$ also moves $(x^{(j)}, y^{(j)}, dir^{(j)})$ to $\Pi_{pub}$.

The above means $(\Pi'_{pub})^{(j)} = \Pi^{(j)}_{pub}$ still holds after the initial action.

Similarly to Case 1 and 2, let $\Pi^{(j-1)}_{tmp} := \Pi^{(j)}_{pub} \cup \Pi^{(j-1)}_{in}$ and $(\Pi'_{tmp})^{(j-1)} := (\Pi'_{pub})^{(j)} \cup (\Pi'_{in})^{(j-1)}$. It still holds $\Pi^{(j-1)}_{tmp} \cong (\Pi'_{tmp})^{(j-1)}$.

**Chain detection and completion.** We consider two types of query structures.

*For every 2-chain $\big((x_2, y_2), (x^{(j)}, y^{(j)})\big)$, $(x_2, y_2) \in \Pi^{(j)}_{pub}$:* Since $(\Pi'_{pub})^{(j)} = \Pi^{(j)}_{pub}$, it also holds $(x_2, y_2) \in (\Pi'_{pub})^{(j)}$. Let $y_1 = x_2 \oplus \varphi^{-1}(y_2 \oplus x^{(j)})$. For the remaining actions, we further distinguish three cases as follows.

- **Case 3.2.a:** $\exists (x_1, y_1) \in \Pi^{(j)}_{pub}$. Since $(\Pi'_{pub})^{(j)} = \Pi^{(j)}_{pub}$, it holds $(x_1, y_1) \in (\Pi'_{pub})^{(j)}$. Then both $\mathcal{T}^{E,\mathbf{P}}$ and $\mathcal{T}^{\mathrm{EMSP}[\varphi]^{\pi}_{4},\pi}$ detect the 3-chain $\big((x_1, y_1), (x_2, y_2), (x^{(j)}, y^{(j)})\big)$. $\mathcal{T}^{E,\mathbf{P}}$ completes $\big((x_1, y_1), (x_2, y_2), (x^{(j)}, y^{(j)}), (x_4, y_4, \perp_{\rightarrow})\big)$, whereas $\mathcal{T}^{\mathrm{EMSP}[\varphi]^{\pi}_{4},\pi}$ completes $\big((x_1, y_1), (x_2, y_2), (x^{(j)}, y^{(j)}), (x_4, y'_4, \perp_{\rightarrow})\big)$ with $y'_4 = \pi(x_4)$. Since $(x^\circ, y^\circ) \in \Pi^{(j-1)}_{all}$, it necessarily holds $x_4 \neq x^\circ$, and thus $y'_4 = y_4$ by $\pi \vdash \big(\Pi^{(\ell)}_{all} \backslash \{(x^\circ, y^\circ)\}\big)$. By this, both $\mathcal{T}^{E,\mathbf{P}}$ and $\mathcal{T}^{\mathrm{EMSP}[\varphi]^{\pi}_{4},\pi}$ creates the record $(x_4, y_4, \perp_{\rightarrow})$, and thus $\Pi'_{in} = \Pi_{in}$ after completing this 4-chain.

- **Case 3.2.b:** $\exists (x_1, y_1, dir_1) \in \Pi^{(j-1)}_{in}$. Then $\mathcal{T}^{E,\mathbf{P}}$ detects $\big((x_1, y_1), (x_2, y_2), (x^{(j)}, y^{(j)})\big)$ and completes $\big((x_1, y_1, dir_1), (x_2, y_2), (x^{(j)}, y^{(j)}), (x_4, y_4, \perp_{\rightarrow})\big)$. It has to be $dir_1 \in \{\leftarrow, \perp_{\leftarrow}\}$: otherwise, $B$ querying $x^{(j)} = y_2 \oplus \varphi(y_1 \oplus x_2)$ would have caused $B^{\mathcal{T}^{E,\mathbf{P}}}$ abort at line 39 in *CheckPrivacy*$(x^{(j)})$. By this and by $\Pi^{(j-1)}_{all} \cong (\Pi'_{all})^{(j-1)}$, it holds $(x'_1, y_1, dir_1) \in \Pi^{(j-1)}_{in}$ as well, where $x'_1 = \pi^{-1}(y_1)$. Regardless of $x'_1 = x_1$ or not, $\mathcal{T}^{\mathrm{EMSP}[\varphi]^{\pi}_{4},\pi}$ will also detect a 3-chain $\big((x'_1, y_1), (x_2, y_2), (x^{(j)}, y^{(j)})\big)$ and complete $\big((x'_1, y_1), (x_2, y_2), (x^{(j)}, y^{(j)}), (x_4, y'_4, \perp_{\rightarrow})\big)$, where $y'_4 = \mathrm{EMSP}[\varphi]^{\pi}_{4}\big(k, k \oplus x'_1\big) = \pi(x_4)$ $(k = \varphi^{-1}(y_1 \oplus x_2))$.

  Now, since $(x^\circ, y^\circ) \in \Pi^{(j-1)}_{all}$, it necessarily holds $x_4 \neq x^\circ$, and thus $y'_4 = y_4$ by $\pi \vdash \big(\Pi^{(\ell)}_{all} \backslash \{(x^\circ, y^\circ)\}\big)$. By this, $\Pi'_{in} = \Pi_{in}$ after completing this 4-chain.

- **Case 3.2.c:** $y_1 \notin range\big(\Pi^{(j-1)}_{tmp}\big)$. In this case, in $D^{\mathrm{EMSP}[\varphi]^{\mathcal{T}^{E,\mathbf{P}}}_{4},\mathcal{T}^{E,\mathbf{P}}}$ $\mathcal{T}^{E,\mathbf{P}}$ detects $\big((x_2, y_2), (x^{(j)}, y^{(j)})\big)$ and completes $\big((x_1, y_1, \perp_{\leftarrow}), (x_2, y_2), (x^{(j)}, y^{(j)}), (x_4, y_4, \rightarrow)\big)$.

  We argue that it also holds $y_1 \notin range\big((\Pi'_{tmp})^{(j-1)}\big)$. Clearly, it cannot be $y_1 \notin range\big((\Pi'_{pub})^{(j)}\big)$. Then, if $(x'_1, y_1, dir_1) \in (\Pi'_{in})^{(j-1)}$ for $x'_1 = \pi^{-1}(y_1)$, it has to be $dir_1 \in \{\leftarrow, \perp_{\leftarrow}\}$: otherwise, $B$ querying $x^{(j)} = y_2 \oplus \varphi(y_1 \oplus x_2)$ would have caused $B^{\mathcal{T}^{E,\mathbf{P}}}$ abort at line 39 in *CheckPrivacy*$(x^{(j)})$. By this and by $\Pi^{(j-1)}_{all} \cong (\Pi'_{all})^{(j-1)}$, there exists $(x''_1, y_1, dir_1) \in \Pi^{(j-1)}_{in}$ for some $x''_1$, and this contradicts the assumption that $y_1 \notin range\big((\Pi'_{tmp})^{(j-1)}\big)$.

  Therefore, $y_1 \notin range\big((\Pi'_{tmp})^{(j-1)}\big)$ as well, and $\mathcal{T}^{\mathrm{EMSP}[\varphi]^{\pi}_{4},\pi}$ also detects the 2-chain $\big((x_2, y_2), (x^{(j)}, y^{(j)})\big)$ and complete $\big((x'_1, y_1, \perp_{\leftarrow}), (x_2, y_2), (x^{(j)}, y^{(j)}), (x_4, y'_4, \rightarrow)\big)$ with $x'_1 = \pi^{-1}(y_1)$ and $y'_4 = \pi(x_4)$. Since $(x^\circ, y^\circ) \in \Pi^{(j-1)}_{all}$, it holds $y_1 \neq y^\circ$ and $x_4 \neq x^\circ$, and thus $x'_1 = x_1$ and $y'_4 = y_4$ by $\pi \vdash \big(\Pi^{(\ell)}_{all} \backslash \{(x^\circ, y^\circ)\}\big)$. By this, both $\mathcal{T}^{E,\mathbf{P}}$ and $\mathcal{T}^{\mathrm{EMSP}[\varphi]^{\pi}_{4},\pi}$ create $(x_1, y_1, \perp_{\leftarrow})$ and $(x_4, y_4, \rightarrow)$, and thus $\Pi_{in} \cong \Pi'_{in}$ after completing this 4-chain.

*For every 2-chain $\big((x^{(j)}, y^{(j)}), (x_3, y_3)\big)$, $(x_3, y_3) \in \Pi^{(j)}_{pub}$:* It resembles Case 3.2.c: $\mathcal{T}^{E,\mathbf{P}}$ detects $\big((x^{(j)}, y^{(j)}), (x_3, y_3)\big)$ and completes $\big((x_1, y_1, \leftarrow), (x^{(j)}, y^{(j)}), (x_3, y_3), (x_4, y_4, \perp_{\rightarrow})\big)$, and $\mathcal{T}^{\mathrm{EMSP}[\varphi]^{\pi}_{4},\pi}$ detects $\big((x^{(j)}, y^{(j)}), (x_3, y_3)\big)$ and completes $\big((x'_1, y_1, \leftarrow), (x^{(j)}, y^{(j)}), (x_3, y_3),$

$(x_4, y'_4, \bot_{\rightarrow})\big)$ with $x'_1 = \pi^{-1}(y_1)$ and $y'_4 = \pi(x_4)$. Since $(x^\circ, y^\circ) \in \Pi_{all}^{(j-1)}$, it holds $y_1 \neq y^\circ$ and $x_4 \neq x^\circ$, thus $x'_1 = x_1$ and $y'_4 = y_4$ by $\pi \vdash \big(\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}\big)$. Hence, both $\mathcal{T}^{E,\mathbf{p}}$ and $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}$ creates $(x_1, y_1, \leftarrow)$ and $(x_4, y_4, \bot_{\rightarrow})$, and $\Pi'_{in} \cong \Pi_{in}$ after this.

**In summary,** if: (i) $(\Pi'_{all})^{(j-1)} \cong \Pi_{all}^{(j-1)}$, and (ii) $(x^\circ, y^\circ) \in \Pi_{all}^{(j-1)}$ (and thus $(x^\circ, y^\circ) \in \Pi_{all}^{(j)}$ as well), then in the subsequent $j$-th simulator cycle, (i) both $\mathcal{T}^{E,\mathbf{p}}$ and $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}$ add the record $(x^{(j)}, y^{(j)})$ to their sets $\Pi_{pub}$ and $\Pi'_{pub}$ resp., and (ii) every time $\mathcal{T}^{E,\mathbf{p}}$ adds a record $(x, y, dir)$ to its set $\Pi_{in}$, $\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}$ adds the same record $(x, y, dir)$ to its set $\Pi'_{in}$. The above mean $(\Pi'_{all})^{(j)} \cong \Pi_{all}^{(j)}$ after the $j$-th cycle.

### 6.1.4 "Isomorphicness" of $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}}$ and $B^{\mathcal{T}^{E,\mathbf{p}}}$: Concluding

The above means $(\Pi'_{all})^{(j-1)} \cong \Pi_{all}^{(j-1)} \Longrightarrow (\Pi'_{all})^{(j)} \cong \Pi_{all}^{(j)}$. Obviously, $\Pi_{all}^{(0)} \cong (\Pi'_{all})^{(0)}$, since they are both empty. Thus, $(\Pi'_{all})^{(j)} \cong \Pi_{all}^{(j)}$ for all $j \in \{1, ..., \ell\}$. This means:

- $(\Pi'_{pub})^{(j)} = \Pi_{pub}^{(j)}$ for all $j \in \{1, ..., \ell\}$. This means for $j = 1, ..., \ell$, $B$ gets the same response for its $j$-th query in the two executions $B^{\mathcal{T}^{E,\mathbf{p}}}$ and $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}}$. Since $B$ is deterministic, its $(j+1)$-th query in $B^{\mathcal{T}^{E,\mathbf{p}}}$ and $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}}$ are identical. $B$ thus gets the same transcript of queries and responses in $B^{\mathcal{T}^{E,\mathbf{p}}}$ and $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}}$;
- $(\Pi'_{in})^{(j)} \cong \Pi_{in}^{(j)}$ for all $j \in \{1, ..., \ell\}$. Thus, when $(x^\circ, y^\circ, dir^\circ) \in \Pi_{in}^{(\ell)}$, $dir^\circ \in \{\rightarrow, \bot_{\rightarrow}\}$, it holds $(x^\circ, y^{\circ\circ}, dir^\circ) \in \Pi'_{in}$ correspondingly, where $y^{\circ\circ} = \pi(x^\circ)$; when $(x^\circ, y^\circ, dir^\circ) \in \Pi_{in}^{(\ell)}$, $dir^\circ \in \{\leftarrow, \bot_{\leftarrow}\}$, it holds $(x^{\circ\circ}, y^\circ, dir^\circ) \in \Pi'_{in}$ correspondingly, where $x^{\circ\circ} = \pi^{-1}(y^\circ)$.

**Distribution of "private" endpoints.** Fix $(x^\circ, y^\circ, dir^\circ) \in \Pi_{in}^{(\ell)}$, and wlog assume $dir^\circ \in \{\rightarrow, \bot_{\rightarrow}\}$. If all $\pi \vdash \big(\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}\big)$ are good (i.e., execution $B^{\mathcal{T}^{\mathrm{EMSP}[\varphi]_4^{\pi},\pi}}$ never aborts when $\pi \vdash \big(\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}\big)$), then conditioned on the $\big|\Pi_{all}^{(\ell)}\big| - 1$ records in $\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}$, the possible number of "non-aborting" choices for $y^\circ$ is $2^n - \big|\Pi_{all}^{(\ell)}\big| + 1 \geq 2^n - 3q^2$ (using Lemma 3). But the situation is not that ideal: by the fact that $D^{\mathrm{EMSP}[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}},\mathcal{T}^{E,\mathbf{p}}}$ did not abort, $B = D^{\mathrm{EMSP}[\varphi]_4}$ can exclude many values from the set $\{0,1\}^n \backslash range\big(\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}\big)$. To see how many values can be excluded, let $\varepsilon := \Pr_{E,\mathbf{p}}\big[D^{\mathrm{EMSP}[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}},\mathcal{T}^{E,\mathbf{p}}} \text{ aborts}\big]$. Then, the number of values $y^{\circ\circ} \in \{0,1\}^n \backslash range\big(\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}\big)$ that can cause $B^{\mathcal{T}^{E,\mathbf{p}}}$ abort cannot be larger than $2^n \cdot \varepsilon$: otherwise, $B^{\mathcal{T}^{E,\mathbf{p}}}$ aborts with probability $> 2^n \cdot \varepsilon / 2^n = \varepsilon$ when creating the record $(x^\circ, y^{\circ\circ}, dir^\circ)$, and an obvious contradiction with our assumption $\varepsilon \leq 1/2$ is reached. Therefore, the number of "non-aborting" values $y^{\circ\circ} \in \{0,1\}^n \backslash range\big(\Pi_{all}^{(\ell)} \backslash \{(x^\circ, y^\circ)\}\big)$ is at least $2^n - 3q^2 - 2^n \cdot \varepsilon \geq 2^n/2 - 3q^2$.

## 6.2 Abort Probability of $CheckPrivacy$

Consider a call to $CheckPrivacy(x)$, and we analyze the abort conditions in turn.

**Condition at Line 37.** When $B$ issues the query $P(x)$, for any $(x', y', dir') \in \Pi_{in}$ such that $dir' \in \{\leftarrow, \bot_{\leftarrow}\}$, the left endpoint $x'$ is uniform in at least $2^n/2 - 3q^2$ choices (as argued). Assuming $6q^2 \leq 2^n/2$, it holds $\Pr[x = x'] \leq 1/(2^n/2 - 3q^2) \leq 2/(2^n - 6q^2) \leq 4/2^n$. By Lemma 3, the number of internal records is $|\Pi_{in}| \leq 2q^2$. Thus, the probability that a call to $CheckPrivacy(x)$ aborts at line 37 is at most $8q^2/2^n$.

**Condition at Line 39.** For any pair $\big((x_1, y_1, dir_1), (x_2, y_2, dir_2)\big)$, it has three cases.

*Case 1: $(x_2, y_2, dir_2) \in \Pi_{in}$ and $(x_1, y_1) \neq (x_2, y_2)$.* Then, when $B$ issues the query $P(x)$, either $x_2$ or $y_2$ is uniform in at least $2^n/2 - 3q^2$ choices (as argued). By this, the probability to have $x = y_2 \oplus \varphi(y_1 \oplus x_2)$ in this case is bounded by $\frac{1}{2^n/2-3q^2} \leq 4/2^n$.

*Case 2: $(x_2, y_2, dir_2) \in \Pi_{in}$ and $(x_1, y_1) = (x_2, y_2)$.* Then, if $dir_1 \in \{\leftarrow, \perp_\leftarrow\}$, then $x_1$ is uniform in $\geq 2^n/2 - 3q^2$ choices when $B$ queries $P(x)$, and $\Pr[x = y_1 \oplus \varphi(y_1 \oplus x_1)] \leq \frac{1}{2^n/2-3q^2} \leq 4/2^n$. If $dir_1 \in \{\rightarrow, \perp_\rightarrow\}$, then $y_1$ is uniform in $\geq 2^n/2 - 3q^2$ choices when $B$ queries $P(x)$, and $\Pr[x = y_1 \oplus \varphi(y_1 \oplus x_1)] \leq \frac{C(\varphi)}{2^n/2-3q^2} \leq 4C(\varphi)/2^n$.

Thus, it always holds $\Pr[x = y_1 \oplus \varphi(y_1 \oplus x_1)] \leq 4C(\varphi)/2^n$ in this case.

*Case 3: $(x_2, y_2, dir_2) \in \Pi_{pub}$, $(x_1, y_1, dir_1) \in \Pi_{in}$ and $dir_1 \in \{\rightarrow, \perp_\rightarrow\}$.* When $B$ queries $P(x)$, $y_1$ is uniform in $\geq 2^n/2 - 3q^2$ choices. By this, when $(x_1, y_1) \neq (x_2, y_2)$, it holds $\Pr[x = y_2 \oplus \varphi(y_1 \oplus x_2)] \leq \frac{1}{2^n/2-3q^2} \leq 4/2^n$; when $(x_1, y_1) = (x_2, y_2)$, it holds $\Pr[x = y_1 \oplus \varphi(y_1 \oplus x_1)] \leq \frac{C(\varphi)}{2^n/2-3q^2} \leq 4C(\varphi)/2^n$.

By the above, for every pair $\big((x_1, y_1), (x_2, y_2)\big)$, it holds $\Pr[x = y_2 \oplus \varphi(y_1 \oplus x_2)] \leq 4C(\varphi)/2^n$. Summing over the at most $3q^2 \times 3q^2$ choices of $\big((x_1, y_1), (x_2, y_2)\big)$, the probability that a call to $CheckPrivacy(x)$ aborts at line 39 is at most $36C(\varphi)q^4/2^n$.

**Condition at Line 41.** Consider any triple $\big((x_1, y_1, dir_1), (x_2, y_2, dir_2), (x_3, y_3, dir_3)\big)$. To have $x_1 \oplus \varphi^{-1}(y_1 \oplus x_2) = x_3 \oplus \varphi^{-1}(y_3 \oplus x)$, it cannot be $(x_1, y_1) = (x_3, y_3)$.

*Case 1: $(x_1, y_1) \in \Pi_{in}$.* It further consists of two subcases.

- Subcase 1.1: $(x_1, y_1, dir_1) \neq (x_2, y_2, dir_2)$. Then, when $B$ queries $P(x)$, either $x_1$ or $y_1$ is uniform, and thus $\Pr[x_1 \oplus \varphi^{-1}(y_1 \oplus x_2) = x_3 \oplus \varphi^{-1}(y_3 \oplus x)] \leq \frac{1}{2^n/2-3q^2} \leq 4/2^n$.
- Subcase 1.2: $(x_1, y_1, dir_1) = (x_2, y_2, dir_2)$. Similarly to Case 2 of the condition at line 39, it holds $\Pr[x_1 \oplus \varphi^{-1}(y_1 \oplus x_1) = x_3 \oplus \varphi^{-1}(y_3 \oplus x)] \leq \frac{C(\varphi)}{2^n/2-3q^2} \leq 4C(\varphi)/2^n$.

*Case 2: $(x_2, y_2, dir_2) \in \Pi_{in}$ and $dir_2 \in \{\leftarrow, \perp_\leftarrow\}$.* It consists of three subcases.

- Subcase 2.1: $(x_2, y_2) \neq (x_1, y_1)$ and $(x_2, y_2) \neq (x_3, y_3)$. Then, when $B$ queries $P(x)$, $x_2$ is uniform, and $\Pr[x_1 \oplus \varphi^{-1}(y_1 \oplus x_2) = x_3 \oplus \varphi^{-1}(y_3 \oplus x)] \leq \frac{1}{2^n/2-3q^2} \leq 4/2^n$.
- Subcase 2.2: $(x_2, y_2) = (x_1, y_1)$. Since $(x_1, y_1) \neq (x_3, y_3)$, this means $(x_2, y_2) \neq (x_3, y_3)$. Thus $\Pr[x_2 \oplus \varphi^{-1}(y_2 \oplus x_2) = x_3 \oplus \varphi^{-1}(y_3 \oplus x)] \leq \frac{C(\varphi)}{2^n/2-3q^2} \leq 4C(\varphi)/2^n$.
- Subcase 2.3: $(x_3, y_3) = (x_2, y_2)$. Since $(x_1, y_1) \neq (x_3, y_3)$, this means $(x_2, y_2) \neq (x_1, y_1)$, and thus $\Pr[x_1 \oplus \varphi^{-1}(y_1 \oplus x_2) = x_2 \oplus \varphi^{-1}(y_2 \oplus x)] \leq \frac{C(\varphi)}{2^n/2-3q^2} \leq 4C(\varphi)/2^n$.

*Case 3: $(x_3, y_3) \in \Pi_{in}$.* The analysis resembles Case 1 and yields bound $4C(\varphi)/2^n$.

Therefore, in any case, it holds $\Pr[x_1 \oplus \varphi^{-1}(y_1 \oplus x_2) = x_3 \oplus \varphi^{-1}(y_3 \oplus x)] \leq 4C(\varphi)/2^n$. Summing over the at most $(3q^2)^3$ choices of $\big((x_1, y_1), (x_2, y_2), (x_3, y_3)\big)$, the probability that a call to $CheckPrivacy(x)$ aborts at line 41 is at most $108C(\varphi)q^6/2^n$.

### 6.2.1 Summarizing

By union bound, a call $CheckPrivacy(x)$ aborts with probability $\leq 8q^2/2^n + 36C(\varphi)q^4/2^n + 108C(\varphi)q^6/2^n$. The same bound holds for $CheckPrivacy^{-1}(y)$ by symmetry. The total number of $CheckPrivacy(x)$ and $CheckPrivacy^{-1}(y)$ calls is at most $q$. Thus,

$$\Pr\big[CheckPrivacy \text{ and } CheckPrivacy^{-1} \text{ abort}\big] \leq \frac{8q^3}{2^n} + \frac{36C(\varphi)q^5}{2^n} + \frac{108C(\varphi)q^7}{2^n}. \quad (3)$$

## 6.3 Abort Probability of $CheckInternalColl$

Consider a call to $CheckInternalColl(x, y)$ which is only made in a call to $P(x)$. We analyze the abort conditions in turn.

**Condition at Line 50.** When $B$ queries $P(x)$, for any $(x', y', dir') \in \Pi_{in}$ such that $dir' \in \{\rightarrow, \perp_{\rightarrow}\}$, $y'$ is uniform in $\geq 2^n/2 - 3q^2$ choices. This also includes the record $(x, y, dir)$ corresponding to $P(x)$. Since no "influential" actions happen in the call to $P(x)$, the uniformness of $y$ remains till $\mathcal{T}$ checking line 50. Now, consider two cases:

- Case 1: $x_2 = x$. Then $\Pr[x_3 = y \oplus \varphi(y \oplus x)] \leq C(\varphi)/(2^n/2 - q) \leq 4C(\varphi)/2^n$;
- Case 2: $x_2 \neq x$. Then $\Pr[x_3 = y_2 \oplus \varphi(y \oplus x_2)] \leq 1/(2^n/2 - q) \leq 4/2^n$.

By these, $\Pr[x_3 = y_2 \oplus \varphi(y \oplus x_2)] \leq 4C(\varphi)/2^n$ for every $\big((x_2, y_2), (x_3, y_3)\big)$ (assuming $2q \leq 2^n/2$). By Lemma 3, the number of choices for $\big((x_2, y_2), (x_3, y_3)\big)$ is $\leq 9q^4$. Thus, the probability that a call to $CheckInternalColl(x, y)$ aborts at line 50 is $\leq 36C(\varphi)q^4/2^n$.

**Condition at Line 52.** As argued, for the record $(x, y, dir)$ corresponding to $P(x)$, $y$ remains uniform in at least $2^n/2 - 3q^2$ choices until $\mathcal{T}$ checking line 52.

It cannot be $x = x_3$: otherwise, $x \oplus \varphi^{-1}(y \oplus x_2) = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4)$ is not possible, while $y_2 \oplus \varphi(y \oplus x_2) = y_4 \oplus \varphi(y_3 \oplus x_4)$ implies $y_2 \oplus \varphi(x_2) = y_4 \oplus \varphi(x_4)$ and contradicts **Inv1**. Similarly by symmetry, it cannot be $x_2 = x_4$. By these, there remain three cases:

- Case 1: $x_2 = x$, $x_3 \neq x$, $x_4 \neq x$. Then the number of $y'$ s.t. $x \oplus \varphi^{-1}(y' \oplus x) = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4)$ is 1, while the number of $y'$ s.t. $y' \oplus \varphi(y' \oplus x) = y_4 \oplus \varphi(y_3 \oplus x_4)$ is at most $C(\varphi)$;
- Case 2: $x_2 \neq x$, $x_3 \neq x$, $x_4 = x$. Then the number of $y'$ s.t. $x \oplus \varphi^{-1}(y' \oplus x_2) = x_3 \oplus \varphi^{-1}(y_3 \oplus x)$ is 1, while the number of $y'$ s.t. $y_2 \oplus \varphi(y' \oplus x_2) = y' \oplus \varphi(y_3 \oplus x)$ is at most $C(\varphi)$;
- Case 3: $x_2 \neq x$, $x_3 \neq x$, $x_4 \neq x$. Then the number of $y'$ s.t. $x \oplus \varphi^{-1}(y' \oplus x_2) = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4)$ is 1, while the number of $y'$ s.t. $y_2 \oplus \varphi(y' \oplus x_2) = y_4 \oplus \varphi(y_3 \oplus x_4)$ is 1.

By these, for each triple $\big((x_2, y_2), (x_3, y_3), (x_4, y_4)\big)$, $\Pr[x \oplus \varphi^{-1}(y \oplus x_2) = x_3 \oplus \varphi^{-1}(y_3 \oplus x_4) \vee y_2 \oplus \varphi(y \oplus x_2) = y_4 \oplus \varphi(y_3 \oplus x_4)] \leq 2C(\varphi)/(2^n/2 - q) \leq 8C(\varphi)/2^n$ (assuming $2q \leq 2^n/2$). By Lemma 3, $\big((x_2, y_2), (x_3, y_3), (x_4, y_4)\big)$ has at most $27q^6$ choices. Thus, the probability that a call $CheckInternalColl(x, y)$ aborts at line 52 is $\leq 216C(\varphi)q^6/2^n$.

**Condition at Line 54.** As argued, for the record $(x, y, dir)$ corresponding to $P(x)$, $y$ remains uniform in at least $2^n/2 - 3q^2$ choices till $\mathcal{T}$ checking the condition at line 54. It cannot be $x_1 = x_3$: otherwise, $y \oplus \varphi(y_1 \oplus x) = y_4 \oplus \varphi(y_3 \oplus x_4)$ implies $y \oplus \varphi(x) = y_4 \oplus \varphi(x_4)$ and contradicts **Inv1**. It cannot be $x = x_4$ either, as otherwise $y \oplus \varphi(y_1 \oplus x) = y_4 \oplus \varphi(y_3 \oplus x_4)$ is not possible. By these, there remain three possible cases, i.e., $x_1 = x$, $x_3 \neq x$, $x_4 \neq x$; $x_1 \neq x$, $x_3 = x$, $x_4 \neq x$; and $x_1 \neq x$, $x_3 \neq x$, $x_4 \neq x$. In each case, the number of $y'$ such that $y' \oplus \varphi(y' \oplus x) = y_4 \oplus \varphi(y_3 \oplus x_4)$ is at most $C(\varphi)$. By these, $\Pr[y \oplus \varphi(y_1 \oplus x) = y_4 \oplus \varphi(y_3 \oplus x_4)] \leq C(\varphi)/(2^n/2 - q) \leq 4C(\varphi)/2^n$ for each triple $\big((x_1, y_1), (x_3, y_3), (x_4, y_4)\big)$ (assuming $2q \leq 2^n/2$). By Lemma 3, $\big((x_1, y_1), (x_3, y_3), (x_4, y_4)\big)$ have at most $27q^6$ choices, and thus a call $CheckInternalColl(x, y)$ aborts at line 54 with probability $\leq 108C(\varphi)q^6/2^n$.

**Summarizing.** Summing over the above, a call to $\mathit{CheckInternalColl}(x, y)$ aborts with probability $\le 36C(\varphi)q^4/2^n + 216C(\varphi)q^6/2^n + 108C(\varphi)q^6/2^n$. The same bound holds for $\mathit{CheckInternalColl}^{-1}(x, y)$ by symmetry. Clearly, procedures $\mathit{CheckInternalColl}(x, y)$ and $\mathit{CheckInternalColl}^{-1}(x, y)$ are called at most $q$ times in total. Therefore,

$$\Pr\left[\mathit{CheckInternalColl} \text{ and } \mathit{CheckInternalColl}^{-1} \text{ abort}\right] \le \frac{36C(\varphi)q^5 + 324C(\varphi)q^7}{2^n}. \tag{4}$$

### 6.4 Abort Probability of *CheckInterV3Chain*

When $B$ queries $P(x)$, for any $(x', y', dir') \in \Pi_{in}$ such that $dir' \in \{\rightarrow, \perp_{\rightarrow}\}$, $y'$ is uniform in $\ge 2^n/2 - 3q^2$ choices (as argued). This also includes the record $(x, y, dir)$ corresponding to $P(x)$. Since no "influential" actions happen in the call to $P(x)$, the uniformness of $y$ remains till $\mathcal{T}$ makes the call to $\mathit{CheckInterV3Chain}(x', y')$. Similarly, when $dir' \in \{\leftarrow, \perp_{\leftarrow}\}$, $x'$ is uniform in $\ge 2^n/2 - 3q^2$ choices. Therefore, for each choice of four records $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) \in \Pi_{pub}$, $\Pr[\varphi(y' \oplus x) = y \oplus x''] \le 1/(2^n/2 - 3q^2) \le 4/2^n$ for $y' = x_1 \oplus \varphi^{-1}(y_1 \oplus x_2)$ and $x'' = y_4 \oplus \varphi(y_3 \oplus x_4)$. Since the number of such choices is at most $q^4$, and since $\mathit{CheckInterV3Chain}$ is called at most $q \times |\Pi_{in}| \le 3q^3$ times, the probability that $\mathcal{T}^{E,\mathbf{P}}$ aborts in $\mathit{CheckInterV3Chain}$ is at most $\frac{12q^7}{2^n}$.

## 7 Abort Probability of *CheckRecord*

The analysis of the probability that $\mathcal{T}^{E,\mathbf{P}}$ aborts during invocations of the procedure $\mathit{CheckRecord}$ consists of complicated case studies, and we thus spend a whole section. We first exhibit a lemma establishing the (intuitive) quasi-randomness in adapted records.

**Lemma 5.** *In any $\Sigma_2$ execution, right before $\mathcal{T}^{E,\mathbf{P}}$ making a query to $E(k, u)$ ($E^{-1}(k, v)$, resp.), there is no record of the form $(k, u, \star)$ ($(k, \star, v)$, resp.) in $ET$.*

*Proof.* Wlog, consider the case $\mathcal{T}^{E,\mathbf{P}}$ making a forward query $E(k, u)$. By the pseudocode (Sect. 3), $\mathcal{T}^{E,\mathbf{P}}$ only queries $E(k, u)$ in calls to $\mathit{Complete}^+(y_3, k)$. Assume that $\mathit{Complete}^+(y_3, k)$ is called in the $\ell$-th simulator cycle. Furthermore, it can be seen that:

- Right before the call $\mathit{Complete}^+(y_3, k)$, there is a 3-chain $\big((x_1, y_1), (x_2, y_2), (x_3, y_3)\big)$ such that $x_1 = k \oplus u$;
- The $\ell$-th cycle was due to $D$ querying $P(x_2), P^{-1}(y_2), P(x_3)$ or $P^{-1}(y_3)$.

Now, assume that it holds $(k, u, v) \in ET$ for some $v \in \{0, 1\}^n$ before $\mathcal{T}^{E,\mathbf{P}}$ queries $E(k, u)$ in $\mathit{Complete}^+(y_3, k)$. In $\Sigma_2$, it has to be that $\mathcal{T}^{E,\mathbf{P}}$ queried $E(k, u) \rightarrow v$ or $E^{-1}(k, v) \rightarrow u$ in a previous (non-aborting) call to $\mathit{Complete}^+/\mathit{Complete}^-$. Assume that $\mathcal{T}^{E,\mathbf{P}}$ queried $E(k, u) \rightarrow v$ in a previous call to $\mathit{Complete}^+(y_3', k)$. By construction of $\mathrm{EMSP}[\varphi]_4$ and $\mathcal{T}^{E,\mathbf{P}}$, after this call to $\mathit{Complete}^+(y_3', k)$ returns, a corresponding 4-chain $\big((x_1', y_1'), (x_2', y_2'), (x_3', y_3'), (x_4', y_4')\big)$ with $x_1' = k \oplus u$ and $y_4' = \varphi^4(k) \oplus v$ has been in $\Pi_{all}$. Moreover, it holds $(x_2', y_2'), (x_3', y_3') \in \Pi_{pub}$ by Proposition 1. Since $\mathcal{T}^{E,\mathbf{P}}$ did not abort till the (later) call to $\mathit{Complete}^+(y_3, k)$, it can be seen: $x_1' = k \oplus u = x_1$, $y_1' = \Pi_{all}(x_1') = \Pi_{all}(x_1) = y_1$, and further $(x_2', y_2') = (x_2, y_2)$ and $(x_3, y_3) = (x_3', y_3')$. But then $(x_2, y_2), (x_3, y_3) \in \Pi_{pub}$ after the earlier call to $\mathit{Complete}^+(y_3', k)$, and the subsequent $\ell$-th simulator cycle due to $D$ querying $P(x_2), P^{-1}(y_2), P(x_3)$ or $P^{-1}(y_3)$ won't detect and complete chains at all. We thus reach a contradiction.

The case where $\mathcal{T}^{E,\mathbf{P}}$ queried $E^{-1}(k, v) \rightarrow u$ in a previous call to $\mathit{Complete}^-(x_2', k)$ is similar. By the above, the claim holds for the forward query to $E(k, u)$. □

The formal claim is then as follows.

**Lemma 6.** *The probability that $\mathcal{T}^{E,\mathbf{p}}$ aborts inside the procedure CheckRecord is at most* $\big(258C(\varphi)q^6 + 1332q^{10}\big)/2^n$.

To prove Lemma 6, we analyze the conditions in $\mathit{CheckRecord}(x, y, \mathit{dir}, \mathit{num})$ in turn.

## 7.1 Condition at Line 121

Consider the new record $(x, y, \mathit{dir}, \mathit{num})$. First, if $\mathit{dir} =\rightarrow$, then $y = \mathbf{p}(x)$ is uniform in at least $2^n - |\Pi_{all}| \geq 2^n - 3q^2$ choices. Thus, $\Pr[y \oplus \varphi(x) = y' \oplus \varphi(x')] \leq 1/(2^n - 3q^2)$ for each $(x', y')$. The bound $1/(2^n - 3q^2)$ for $\mathit{dir} =\leftarrow$ follows similarly by symmetry.

Second, if $\mathit{dir} = \perp_\rightarrow$, then by Lemma 5, a corresponding ideal cipher query $E(k, u) \rightarrow v$ with $v = y \oplus \varphi^4(k)$ just happened before $\mathit{CheckRecord}(x, y, \perp_\rightarrow, \mathit{num})$. Thus, $y = v \oplus \varphi^4(k)$ is uniform in at least $2^n - q^2$ choices, and $\Pr[y \oplus \varphi(x) = y' \oplus \varphi(x')] \leq 1/(2^n - q^2)$ for each $(x', y')$. The analysis and bound $1/(2^n - q^2)$ for $\mathit{dir} = \perp_\leftarrow$ is similar by symmetry.

Thus, in any case, $\Pr[y \oplus \varphi(x) = y' \oplus \varphi(x')] \leq 1/(2^n - 3q^2)$ for a fixed $(x', y') \in \Pi_{all}$. Since the number of choices for $(x', y')$ is at most $3q^2$, and since *CheckRecord* is called at most $3q^2$ times, the probability that $\mathcal{T}^{E,\mathbf{p}}$ aborts at Line 121 is at most $\frac{(3q^2)^2}{2^n - q^2}$.

## 7.2 Condition at Line 124

As argued in Sect. 7.1, if $\mathit{dir} =\rightarrow$ or $\perp_\rightarrow$, then $y$ is uniform in at least $2^n - 3q^2$ choices; if $\mathit{dir} =\leftarrow$ or $\perp_\leftarrow$, then $x$ is uniform in at least $2^n - 3q^2$ choices. Moreover, the obtained $x$ or $y$ is independent of the values in $\Pi_{all}$. Therefore, for each choice of four records $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) \in \Pi_{all}$, the probability to have $\varphi(y' \oplus x) = y \oplus x''$ for $y' = x_1 \oplus \varphi^{-1}(y_1 \oplus x_2)$ and $x'' = y_4 \oplus \varphi(y_3 \oplus x_4)$ is at most $1/(2^n - 3q^2)$. Since the number of such choices is at most $(3q^2)^4$, and since *CheckRecord* is called at most $3q^2$ times, the probability that $\mathcal{T}^{E,\mathbf{p}}$ aborts at Line 124 is at most $\frac{(3q^2)^5}{2^n - q^2}$.

## 7.3 Condition at Line 126

We distinguish subconditions as follows.

**Subcondition 1:** The new record $(x, y, \mathit{dir})$ gives rise to a triple $(x_1, y_1, \mathit{dir}_1, \mathit{num}_1)$, $(x_2, y_2, \mathit{dir}_2, \mathit{num}_2)$, $(x_3, y_3, \mathit{dir}_3, \mathit{num}_3)$ with $\mathit{num}_1 = \mathit{num}_2 = \mathit{num}_3$ that satisfies the condition at Line 126. This means $x_1 = x$, and $y \oplus \varphi(y \oplus x) = x \Leftrightarrow y \oplus \varphi(y) = x \oplus \varphi(x)$. Now, if $\mathit{dir} =\rightarrow$, then $y = \mathbf{p}(x)$ is uniform in $\geq 2^n - 3q^2$ choices. By our assumption on $\varphi$, the number of $y^\circ$ such that $y^\circ \oplus \varphi(y^\circ) = x \oplus \varphi(x)$ is at most $C(\varphi)$. Thus, $\Pr[y \oplus \varphi(y) = x \oplus \varphi(x)] \leq C(\varphi)/(2^n - 3q^2)$. The same bound holds for $\mathit{dir} =\leftarrow$.

If $\mathit{dir} = \perp_\rightarrow$, then as argued in Sect. 7.1, $y$ is uniform in $\geq 2^n - q^2$ choices, and $\Pr[y \oplus \varphi(y) = x \oplus \varphi(x)] \leq C(\varphi)/(2^n - 3q^2)$. The same bound holds for $\mathit{dir} = \perp_\leftarrow$.

The four cases $(\mathit{dir} =\leftarrow, \perp_\leftarrow, \rightarrow, \perp_\rightarrow)$ are mutual exclusive. Thus, the probability that Subcondition 1 is fulfilled w.r.t. $(x, y)$ is at most $C(\varphi)/(2^n - 3q^2)$.

**Subcondition 2:** The new record $(x, y, \mathit{dir})$ gives rise to a triple $(x_1, y_1, \mathit{dir}_1, \mathit{num}_1)$, $(x_2, y_2, \mathit{dir}_2, \mathit{num}_2), (x_3, y_3, \mathit{dir}_3, \mathit{num}_3)$ with $\mathit{num}_1 = \mathit{num}_2 \neq \mathit{num}_3$ that satisfies the condition at Line 126. Then, depending on the role of $(x, y, \mathit{dir})$, we further analyze two subconditions as follows.

*Subcondition 2.1: there exists $(x_3, y_3, \mathit{dir}_3, \mathit{num}_3)$ such that $\mathit{num}_3 < \mathit{num}$,* and $y \oplus \varphi(y \oplus x) = x_3$. Then, for each such record $(x_3, y_3, \mathit{dir}_3, \mathit{num}_3)$, if $\mathit{dir} \in \{\leftarrow, \perp_\leftarrow\}$, then $x$ is uniform in $\geq 2^n - 3q^2$ possibilities (as argued), and $\Pr[y \oplus \varphi(y \oplus x) = x_3] \leq 1/(2^n - 3q^2)$; if $\mathit{dir} \in \{\rightarrow, \perp_\rightarrow\}$, then $y$ is uniform in $\geq 2^n - 3q^2$ possibilities, and $\Pr[y \oplus \varphi(y \oplus x) = x_3] \leq C(\varphi)/(2^n - 3q^2)$. The number of choices for $(x_3, y_3, \mathit{dir}_3, \mathit{num}_3)$ is at most $3q^2$. Thus, the probability that Subcondition 2.1 is fulfilled w.r.t. $(x, y)$ is $\leq 3C(\varphi)q^2/(2^n - 3q^2)$.

*Subcondition 2.2: $dir \in \{\leftarrow, \perp_{\leftarrow}\}$, and there exists $(x_1, y_1, dir_1, num_1)$ such that* $num_1 < num$ and $y_1 \oplus \varphi(y_1 \oplus x_1) = x$. Again, $x$ is uniform in $\geq 2^n - 3q^2$ choices regardless of $dir = \leftarrow$ or $\perp_{\leftarrow}$, and $\Pr[y_1 \oplus \varphi(y_1 \oplus x_1) = x] \leq 1/(2^n - 3q^2)$ for each $(x_1, y_1, dir_1, num_1)$. Thus, the probability that Subcondition 2.2 is fulfilled w.r.t. $(x, y)$ is $\leq 3q^2/(2^n - 3q^2)$.

A union bound over the subconditions yield that the total probability that Subcondition 2 is fulfilled w.r.t. $(x, y)$ is at most $3C(\varphi)q^2/(2^n - 3q^2) + 3q^2/(2^n - 3q^2)$.

**Subcondition 3:** $(x, y)$ yields $(x_1, y_1, dir_1, num_1)$, $(x_2, y_2, dir_2, num_2)$, $(x_3, y_3, dir_3, num_3)$ with $num_1 \neq num_2 = num_3$ that satisfies the condition at Line 126. The case-study is actually similar to Subcondition 2 by symmetry, showing that Subcondition 3 is fulfilled w.r.t. $(x, y)$ with probability at most $3C(\varphi)q^2/(2^n - 3q^2) + 3q^2/(2^n - 3q^2)$.

**Subcondition 4:** The new record $(x, y, dir)$ gives rise to a triple $(x_1, y_1, dir_1, num_1)$, $(x_2, y_2, dir_2, num_2)$, $(x_3, y_3, dir_3, num_3)$ with $num_1 = num_3 \neq num_2$ that satisfies the condition at Line 126. We further analyze two subconditions as follows.

*Subcondition 4.1: there exists $(x_2, y_2, dir_2, num_2)$ such that $num_2 < num$,* and $y_2 \oplus \varphi(y \oplus x_2) = x$. Then, for each $(x_2, y_2, dir_2, num_2)$, regardless of the value of $dir$, either $x$ or $y$ is uniform in $\geq 2^n - 3q^2$ choices (as argued), and $\Pr[y_2 \oplus \varphi(y \oplus x_2) = x] \leq 1/(2^n - 3q^2)$. Summing over the at most $3q^2$ choices of $(x_2, y_2)$, the probability to have Subcondition 4.1 is at most $3q^2/(2^n - 3q^2)$.

*Subcondition 4.2: there exists $(x_1, y_1, dir_1, num_1)$ such that $num_1 < num$,* and $y \oplus \varphi(y_1 \oplus x) = x_1$. Similarly to Subcondition 4.1, either $x$ or $y$ is uniform in $\geq 2^n - 3q^2$ possibilities, and $\Pr[y \oplus \varphi(y_1 \oplus x) = x_1] \leq 1/(2^n - 3q^2)$ for each $(x_1, y_1)$, and further Subcondition 4.2 holds with probability at most $3q^2/(2^n - 3q^2)$.

Thus, Subcondition 4 holds w.r.t. $(x, y)$ with probability $\leq 6q^2/(2^n - 3q^2)$.

**Subcondition 5:** $(x, y, dir)$ gives rise to a triple $(x_1, y_1, dir_1, num_1)$, $(x_2, y_2, dir_2, num_2)$, $(x_3, y_3, dir_3, num_3)$ with distinct $num_1, num_2$ and $num_3$.

*Subcondition 5.1: there exist distinct $(x_2, y_2, dir_2, num_2)$ and $(x_3, y_3, dir_3, num_3)$* such that $num_2, num_3 < num$, $dir \in \{\rightarrow, \perp_{\rightarrow}\}$, and $y_2 \oplus \varphi(y \oplus x_2) = x_3$. The number of choices for $\big((x_2, y_2), (x_3, y_3)\big)$ is at most $3q^2(3q^2-1)$, and the probability for each is at most $1/(2^n-3q^2)$. Thus, the total probability of Subcondition 5.1 is at most $3q^2(3q^2 - 1)/(2^n - 3q^2)$.

*Subcondition 5.2: there exist distinct $(x_1, y_1, dir_1, num_1)$ and $(x_3, y_3, dir_3, num_3)$* such that $num_1, num_3 < num$ and $y \oplus \varphi(y_1 \oplus x) = x_3$. The number of choices for $\big((x_1, y_1), (x_3, y_3)\big)$ is at most $3q^2(3q^2 - 1)$, and the probability for each is at most $1/(2^n - 3q^2)$ regardless of $dir$. Thus, the total probability of Subcondition 5.2 is at most $3q^2(3q^2 - 1)/(2^n - 3q^2)$.

*Subcondition 5.3: there exists distinct $(x_1, y_1, dir_1, num_1)$ and $(x_2, y_2, dir_2, num_2)$* such that $num_1, num_2 < num$, $dir \in \{\leftarrow, \perp_{\leftarrow}\}$, and $y_2 \oplus \varphi(y_1 \oplus x_2) = x$. Again, the total probability of Subcondition 5.3 is at most $3q^2(3q^2 - 1)/(2^n - 3q^2)$.

Thus, Subcondition 5 holds w.r.t. $(x, y)$ with probability $\leq 9q^2(3q^2 - 1)/(2^n - 3q^2)$.

**Summarizing.** Summing over the five subconditions and considering that $CheckRecord$ is called at most $3q^2$ times, the probability that $\mathcal{T}^{E,\mathbf{P}}$ aborts at Line 126 is bounded by

$$3q^2 \times \Big(\frac{C(\varphi)}{2^n - 3q^2} + \frac{6C(\varphi)q^2}{2^n - 3q^2} + \frac{6q^2}{2^n - 3q^2} + \frac{6q^2}{2^n - 3q^2} + \frac{9q^2(3q^2 - 1)}{2^n - 3q^2}\Big) \leq \frac{21C(\varphi)q^4 + 90q^6}{2^n - 3q^2}.$$

## 7.4 Conditions at Lines 128 and 130

We classify the subconditions as follows.

*Type-A:* the new record $(x, y)$ appears 3 times in the two involved 2-chains. Furthermore,

- Subcondition A.1: the record $(x, y, dir, num)$ forms two 2-chains $\big((x, y), (x, y)\big)$ and $\big((x', y', dir', num'), (x, y)\big)$ with $y \oplus \varphi(y \oplus x) = y \oplus \varphi(y' \oplus x)$. This is not possible.
- Subcondition A.2: the record $(x, y, dir, num)$ forms two 2-chains $\big((x, y), (x, y)\big)$ and $\big((x, y), (x', y', dir', num')\big)$ such that $num' < num$ and $y \oplus \varphi(y \oplus x) = y' \oplus \varphi(y \oplus x')$. But this implies $y \oplus \varphi(x) = y' \oplus \varphi(x')$ and earlier abortion at Line 121.

  Other possibilities of **Type-A** conditions are essentially equivalent with A.1 or A.2.

*Type-B:* the new record $(x, y)$ appears twice in the two involved 2-chains, and the other two involved records are *the same*. Furthermore,

- Subcondition B.1: $(x, y, dir, num)$ forms 2-chains $\big((x, y), (x, y)\big)$ and $\big((x', y', dir', num'), (x', y', dir', num')\big)$ such that $num' < num$ and $y \oplus \varphi(y \oplus x) = y' \oplus \varphi(y' \oplus x')$.

  As argued, either $x$ or $y$ is uniform in $\geq 2^n - 3q^2$ choices. Thus, $\Pr[y \oplus \varphi(y \oplus x) = y' \oplus \varphi(y' \oplus x')] \leq 3C(\varphi)q^2/(2^n - 3q^2)$ for one of the $3q^2$ choices of $(x', y')$.
- Subcondition B.2: the record $(x, y, dir, num)$ forms $\big((x, y), (x', y', dir', num')\big)$ and $\big((x', y', dir', num'), (x, y)\big)$ such that $num' < num$ and $y' \oplus \varphi(y \oplus x') = y \oplus \varphi(y' \oplus x)$.

  In a similar vein to Subcondition B.1, the probability to have $y' \oplus \varphi(y \oplus x') = y \oplus \varphi(y' \oplus x)$ for one of the $3q^2$ choices of $(x', y')$ is at most $3C(\varphi)q^2/(2^n - 3q^2)$.

  Other possibilities of **Type-B** conditions are essentially equivalent with B.1 or B.2.

*Type-C:* the new record $(x, y)$ appears twice in the two involved 2-chains, and the other two involved records are *distinct*. Furthermore,

- Subcondition C.1: the record $(x, y, dir, num)$ forms two 2-chains $\big((x, y), (x, y)\big)$ and $\big((x_1, y_1, dir_1, num_1), (x_2, y_2, dir_2, num_2)\big)$ such that $num_1, num_2 < num$ and $y \oplus \varphi(y \oplus x) = y_2 \oplus \varphi(y_1 \oplus x_2)$.

  Regardless of the value of $dir$, either $x$ or $y$ is uniform in $\geq 2^n - 3q^2$ choices. Thus, $\Pr[y \oplus \varphi(y \oplus x) = y_2 \oplus \varphi(y_1 \oplus x_2)] \leq 3C(\varphi)q^2(3q^2 - 1)/(2^n - 3q^2)$ for one of the $3q^2(3q^2 - 1)$ choices of $\big((x_1, y_1), (x_2, y_2)\big)$.
- Subcondition C.2: the record $(x, y, dir, num)$ forms $\big((x, y), (x_1, y_1, dir_1, num_1)\big)$ and $\big((x, y), (x_2, y_2, dir_2, num_2)\big)$ such that $num_1, num_2 < num$ and $y_1 \oplus \varphi(y \oplus x_1) = y_2 \oplus \varphi(y \oplus x_2)$. But this implies $y_1 \oplus \varphi(x_1) = y_2 \oplus \varphi(x_2)$ and $\mathcal{T}^{E,\mathbf{P}}$ should have aborted at Line 121 during creating the later of $(x_1, y_1)$ and $(x_2, y_2)$.
- Subcondition C.3: the record $(x, y, dir, num)$ forms $\big((x, y), (x_1, y_1, dir_1, num_1)\big)$ and $\big((x_2, y_2, dir_2, num_2), (x, y)\big)$ with $num_1, num_2 < num$ and $y_1 \oplus \varphi(y \oplus x_1) = y \oplus \varphi(y_2 \oplus x)$. The bound $3C(\varphi)q^2(3q^2 - 1)/(2^n - 3q^2)$ is similar to Subcondition C.1.
- Subcondition C.4: the record $(x, y, dir, num)$ forms $\big((x_1, y_1, dir_1, num_1), (x, y)\big)$ and $\big((x_2, y_2, dir_2, num_2), (x, y)\big)$ with $y \oplus \varphi(y_1 \oplus x) = y \oplus \varphi(y_2 \oplus x)$. This is not possible.

  Other possibilities of **Type-C** conditions are equivalent with one of the above.

*Type-D:* the new record $(x, y)$ appears once in the two involved 2-chains. Furthermore,

- Subcondition D.1: the record $(x, y, dir, num)$ forms $\big((x, y), (x_2, y_2, dir_2, num_2)\big)$ and $\big((x_3, y_3, dir_3, num_3), (x_4, y_4, dir_4, num_4)\big)$ such that $dir \in \{\rightarrow, \perp_\rightarrow\}$ and $num_2, num_3, num_4 < num$ and $y_2 \oplus \varphi(y \oplus x_2) = y_4 \oplus \varphi(y_3 \oplus x_4)$.

  The number of choices for $(x_2, y_2), (x_3, y_3), (x_4, y_4)$ is at most $(3q^2)^3$, and the probability to have $y_2 \oplus \varphi(y \oplus x_2) = y_4 \oplus \varphi(y_3 \oplus x_4)$ for each is at most $1/(2^n - 3q^2)$. Thus, the total probability of Subcondition D.1 is at most $(3q^2)^3/(2^n - 3q^2)$.

- Subcondition D.2: the record $(x, y, dir, num)$ forms $\big((x_1, y_1, dir_1, num_1), (x, y)\big)$ and $\big((x_3, y_3, dir_3, num_3), (x_4, y_4, dir_4, num_4)\big)$ such that $num_1, num_3, num_4 < num$ and $y \oplus \varphi(y_1 \oplus x) = y_4 \oplus \varphi(y_3 \oplus x_4)$. Similarly to D.1, the total probability of Subcondition D.2 is at most $(3q^2)^3/(2^n - 3q^2)$.

  Other possibilities of **Type-D** conditions are equivalent with one of the above.

*Summarizing.* Summing over the four types of conditions, and since *CheckRecord* is called at most $3q^2$ times, the probability that $\mathcal{T}^{E,\mathbf{P}}$ aborts at Line 128 is bounded by

$$3q^2 \times \Big(\frac{6C(\varphi)q^2}{2^n - 3q^2} + \frac{6C(\varphi)q^2(3q^2-1)}{2^n - 3q^2} + \frac{2(3q^2)^3}{2^n - 3q^2}\Big) \leq \frac{54C(\varphi)q^6 + 162q^8}{2^n - 3q^2}.$$

Analyses of the probabilities that $\mathcal{T}^{E,\mathbf{P}}$ aborts at Line 130 are similar to Sect. 7.4 by symmetry, yielding the same bound $\frac{54C(\varphi)q^6+162q^8}{2^n-3q^2}$.

### 7.5 Finalizing the Analysis of *CheckRecord*

Summing over the bounds from Sect. 7.1, 7.2, 7.3 and 7.4 yields (using $3q^2 \leq 2^n/2$)

$$\Pr\big[\mathit{CheckRecord}\text{ aborts}\big] \;\leq\; \frac{252q^{10}}{2^n - q^2} + \frac{129C(\varphi)q^6 + 414q^8}{2^n - 3q^2} \;\leq\; \frac{258C(\varphi)q^6 + 1332q^{10}}{2^n}. \tag{5}$$

## 8 Abort Probability of Adaptations, and Concluding

To conclude on the abort probability of $\Sigma_2$, it remains to analyze adaptations. To this end, let us first have a quick overview on simulator cycles and introduce *bad records.*

**Bad Records** We first present a quick overview of the processes to gain some central intuitions. Wlog, consider the case of $D$ querying $P(x)$, as the converse case is similar by symmetry. Regardless of whether $x \in domain(\Pi_{in})$, $\mathcal{T}$ adds a corresponding record $(x, y, dir, num)$ to $\Pi_{pub}$. It is expected to have $dir \in \{\rightarrow, \perp_\rightarrow\}$. $\mathcal{T}$ then makes a call to $\mathit{ProcessRecord}(x, y, dir)$ to "process" $(x, y)$. In this call, $\mathcal{T}$ considers a pile of 3-chains and 2-chains. The case of $dir = \rightarrow$ is illustrated in Fig. 2: $\mathcal{T}$ tries to complete them by adding the underlined records to $\Pi_{in}$.

$$\begin{aligned}
&\big((x_1^{(1)}, y_1^{(1)}), (x_2^{(1)}, y_2^{(1)}), (x, y, \rightarrow), \underline{(x_4^{(1)}, y_4^{(1)}, \perp_\rightarrow)}\big),\\
&\ldots,\\
&\big((x_1^{(\alpha)}, y_1^{(\alpha)}), (x_2^{(\alpha)}, y_2^{(\alpha)}), (x, y, \rightarrow), \underline{(x_4^{(\alpha)}, y_4^{(\alpha)}, \perp_\rightarrow)}\big);\\
&\big(\underline{(x_1^{(\alpha+1)}, y_1^{(\alpha+1)}, \perp_\leftarrow)}, (x_2^{(\alpha+1)}, y_2^{(\alpha+1)}), (x, y, \rightarrow), \underline{(x_4^{(\alpha+1)}, y_4^{(\alpha+1)}, \rightarrow)}\big),\\
&\ldots,\\
&\big(\underline{(x_1^{(\alpha+\beta)}, y_1^{(\alpha+\beta)}, \perp_\leftarrow)}, (x_2^{(\alpha+\beta)}, y_2^{(\alpha+\beta)}), (x, y, \rightarrow), \underline{(x_4^{(\alpha+\beta)}, y_4^{(\alpha+\beta)}, \rightarrow)}\big);\\
&\big(\underline{(x_1^{(\alpha+\beta+1)}, y_1^{(\alpha+\beta+1)}, \leftarrow)}, (x, y, \rightarrow), (x_3^{(\alpha+\beta+1)}, y_3^{(\alpha+\beta+1)}), \underline{(x_4^{(\alpha+\beta+1)}, y_4^{(\alpha+\beta+1)}, \perp_\rightarrow)}\big),\\
&\ldots,\\
&\big(\underline{(x_1^{(\alpha+\beta+\gamma)}, y_1^{(\alpha+\beta+\gamma)}, \leftarrow)}, (x, y, \rightarrow), (x_3^{(\alpha+\beta+\gamma)}, y_3^{(\alpha+\beta+\gamma)}), \underline{(x_4^{(\alpha+\beta+\gamma)}, y_4^{(\alpha+\beta+\gamma)}, \perp_\rightarrow)}\big).
\end{aligned}$$

**Figure 2:** 2-chains and 3-chains addressed by a call to $\mathit{ProcessRecord}(x, y, \rightarrow)$, where $\alpha$, $\beta$ and $\gamma$ are sequence numbers.

It can be seen if the record $(x, y)$ has been involved in certain collisions, then subsequent *Adapt*-calls are deemed to abort. We thus characterize such collisions and defined *bad records*. In detail, a record $(x, y, dir)$ with $dir \in \{\rightarrow, \perp_{\rightarrow}\}$ is *bad*, if any of the following conditions is fulfilled:

- (B-1) There exist two records $(x_2, y_2), (x_4, y_4) \in \Pi_{all}$ s.t. $\big((x_2, y_2), (x, y), (x_4, y_4)\big)$ constitutes a 3-chain, i.e., $x_4 = y \oplus \varphi(y_2 \oplus x)$;
- (B-2) There exist two records $(x_3, y_3), (x_4, y_4) \in \Pi_{all}$ s.t. $\big((x, y), (x_3, y_3), (x_4, y_4)\big)$ constitutes a 3-chain, i.e., $x_4 = y_3 \oplus \varphi(y \oplus x_2)$;
- (B-3) There exist two records $(x_2, y_2), (x_3, y_3) \in \Pi_{pub}$ such that the two 2-chains $\big((x_2, y_2), (x, y)\big)$ and $\big((x, y), (x_3, y_3)\big)$ collide on either left or right, i.e.,
  - $x_2 \oplus \varphi^{-1}(y_2 \oplus x) = x \oplus \varphi^{-1}(y \oplus x_3)$; or
  - $y \oplus \varphi(y_2 \oplus x) = y_3 \oplus \varphi(y \oplus x_3)$.
- (B-4) There exist distinct $(x_2, y_2), (x'_2, y'_2) \in \Pi_{all}$ s.t. the two 2-chains $\big((x_2, y_2), (x, y)\big)$ and $\big((x'_2, y'_2), (x, y)\big)$ collide on left, i.e., $x_2 \oplus \varphi^{-1}(y_2 \oplus x) = x'_2 \oplus \varphi^{-1}(y'_2 \oplus x)$;
- (B-5) There exist distinct $(x_3, y_3), (x'_3, y'_3) \in \Pi_{all}$ s.t. the two 2-chains $\big((x, y), (x_3, y_3)\big)$ and $\big((x, y), (x'_3, y'_3)\big)$ collide on right, i.e., $y_3 \oplus \varphi(y \oplus x_3) = y'_3 \oplus \varphi(y \oplus x'_3)$.

Bad record with $dir \in \{\leftarrow, \perp_{\leftarrow}\}$ is defined symmetrically, and is omitted due to space.

**Proof flow.** Our subsequent arguments proceed in two steps. First, in Lemmas 7—10, we prove that $ProcessRecord(x, y, dir)$ is always called for good $(x, y, dir)$. Then, in Lemma 12, we prove that in *ProcessRecord*-calls with good records, adaptions have bounded abort probabilities. This enables concluding on abort probability in Lemma 15.

## 8.1 Unprocessed Records Are Always Good

This section proceeds with steps as follows.

1. Lemma 7 proves every record $(x, y, dir)$ is good right after it is created;
2. Lemma 8 proves $(x, y, dir)$ remains good at the end of the cycle that creates it;
3. Lemma 9 proves $(x, y, dir)$ remains good after a subsequent new simulator cycle;
4. Lemma 10 proves that every record $(x, y, dir)$ remains good after a subsequent transferring simulator cycle. We finally conclude on the goodness in Lemma 11.

**Lemma 7.** *Every record $(x, y, dir)$ is good right after it is added to $\Pi_{all}$.*

*Proof.* Wlog, consider the case $dir \in \{\rightarrow, \perp_{\rightarrow}\}$. Right after $(x, y, dir)$ is created, if (B-1) is fulfilled, then there is a 3-chain $\big((x_2, y_2, dir_2, num_2), (x, y, dir, num), (x_4, y_4, dir_4, num_4)\big)$ such that $num \geq num_2, num_4$. This clearly contradicts **Inv2**. Similarly, if (B-2) is fulfilled, then it contradicts **Inv2**.

Then, if (B-3) holds, then there exist two 2-chains $\big((x_2, y_2), (x, y)\big)$ and $\big((x, y), (x_3, y_3)\big)$ with colliding endpoints. However, since $(x, y)$ has the largest $num$ value, $x_2 \oplus \varphi^{-1}(y_2 \oplus x) = x \oplus \varphi^{-1}(y \oplus x_3)$ contradicts **Inv4**, while $y \oplus \varphi(y_2 \oplus x) = y_3 \oplus \varphi(y \oplus x_3)$ contradicts **Inv3**.

Finally, $x_2 \oplus \varphi^{-1}(y_2 \oplus x) = x'_2 \oplus \varphi^{-1}(y'_2 \oplus x)$ for (B-4) implies $x_2 \oplus \varphi^{-1}(y_2) = x'_2 \oplus \varphi^{-1}(y'_2)$, contradicting **Inv1**; $y_3 \oplus \varphi(y \oplus x_3) = y'_3 \oplus \varphi(y \oplus x'_3)$ for (B-5) implies $y_3 \oplus \varphi(x_3) = y'_3 \oplus \varphi(x'_3)$, contradicting **Inv1**. So neither (B-4) nor (B-5) is possible. □

**Lemma 8.** *Assume that in a simulator cycle due to $D$ querying $P(x^*) \rightarrow y^*$ or $P^{-1}(y^*) \rightarrow x^*$, $\mathcal{T}$ adds a record $(x, y, dir, num)$ to $\Pi_{in}$. Then, $(x, y, dir, num)$ remains good at the end of this simulator cycle.*

*Proof.* By design, subsequently in this cycle, any newly created record $(x^\circ, y^\circ, dir^\circ, num^\circ)$ must have $(x^\circ, y^\circ) \in \Pi_{in}$. Let $(x^*, y^*, dir^*, num^*)$ be the record of the query $P(x^*) \to y^*$ or $P^{-1}(y^*) \to x^*$. Note that $num^\circ > num > num^*$ by assumption. Moreover, for any $(x', y', dir', num') \in \Pi_{pub}$, it holds $num > num'$. Further note that:

- (B-1) cannot be suddenly fulfilled after $(x^\circ, y^\circ, dir^\circ, num^\circ)$ is created: otherwise, there appears a *bad* 3-chain $\big((x_2, y_2), (x, y), (x_4, y_4)\big)$ and contradicts Lemma 2;
- (B-3) cannot be fulfilled either, since $\mathcal{T}$ never creates new public records after $(x, y)$;
- Neither (B-4) nor (B-5) can be fulfilled, since they contradict **Inv1**.

It remains to analyze (B-2). For this, we distinguish two cases depending on $dir$.

**Case 1: $dir \in \{\to, \bot_\to\}$.** If (B-2) is fulfilled, then a 3-chain $\big((x, y), (x_3, y_3), (x_4, y_4)\big)$ suddenly appears after $(x^\circ, y^\circ, dir^\circ, num^\circ)$ is created. We analyze three subcases as follows.

*Subcase 1.1: $(x_3, y_3) = (x^\circ, y^\circ)$* Then since $(x^\circ, y^\circ) \in \Pi_{in}$, $\big((x, y), (x^\circ, y^\circ), (x_4, y_4)\big)$ is a bad 3-chain, and this contradicts Lemma 2.

*Subcase 1.2: $(x_3, y_3) \in \Pi_{pub}$* (thus, $(x_3, y_3) \neq (x^\circ, y^\circ)$), $(x_4, y_4) = (x^\circ, y^\circ)$, and $dir^\circ \in \{\leftarrow, \bot_\leftarrow\}$. Then it contradicts **Inv2**.

*Subcase 1.3: $(x_3, y_3, dir_3, num_3) \in \Pi_{pub}$, $(x_4, y_4) = (x^\circ, y^\circ)$, and $dir^\circ \in \{\to, \bot_\to\}$* Then, by Proposition 2, right before $\mathcal{T}$ creating $(x^\circ, y^\circ)$, there exists $(x', y', dir', num') \in \Pi_{pub}$ s.t.:

- The 2-chain $\big((x', y'), (x^*, y^*)\big)$ has its "right endpoint" collide with the 2-chain $\big((x, y), (x_3, y_3)\big)$, i.e., $y^* \oplus \varphi(y' \oplus x^*) = x^\circ = y_3 \oplus \varphi(y \oplus x_3)$; or
- The 2-chain $\big((x^*, y^*), (x', y')\big)$ has its "right endpoint" collide with the 2-chain $\big((x, y), (x_3, y_3)\big)$, i.e., $y' \oplus \varphi(y^* \oplus x') = x^\circ = y_3 \oplus \varphi(y \oplus x_3)$.

As discussed, $num > num^*, num', num_3$, i.e., $(x, y, dir, num)$ is "latest". Moreover, $dir \in \{\to, \bot_\to\}$ in Case 1. Therefore, both possibilities contradict **Inv3**.

**Case 2: $dir \in \{\leftarrow, \bot_\leftarrow\}$.** In this case, if (B-2) is fulfilled, then it means a 3-chain $\big((x_1, y_1), (x_2, y_2), (x, y)\big)$ suddenly appears after $(x^\circ, y^\circ, dir^\circ, num^\circ)$ is created. The subcases are similar to Case 1 by symmetry: if $(x_2, y_2) = (x^\circ, y^\circ)$ then $\big((x_1, y_1), (x^\circ, y^\circ), (x, y)\big)$ is bad; if $(x_2, y_2) \in \Pi_{pub} \wedge (x_1, y_1) = (x^\circ, y^\circ) \wedge dir^\circ \in \{\to, \bot_\to\}$ then it contradicts **Inv2**; if $(x_2, y_2) \in \Pi_{pub} \wedge (x_1, y_1) = (x^\circ, y^\circ) \wedge dir^\circ \in \{\leftarrow, \bot_\leftarrow\}$ then we have two 2-chains collide "at the left", contradicting **Inv4**. Thus the claim. □

**Lemma 9.** *Assume that when $D$ queries $P(x^*) \to y^*$ ($P^{-1}(y^*) \to x^*$, resp.), it holds $x^* \notin domain(\Pi_{all})$ ($y^* \notin range(\Pi_{all})$, resp.), and there exists a record $(x, y, dir, num) \in \Pi_{in}$ that is good. Then, after the simulator cycle due to $D$ querying $P(x^*) \to y^*$ or $P^{-1}(y^*) \to x^*$, $(x, y, dir, num)$ remains good.*

*Proof.* For clarity, we list the query records that are relevant to the analysis:

- The record $(x, y, dir, num)$ created before $D$ querying $P(x^*) \to y^*$ or $P^{-1}(y^*) \to x^*$;
- The record $(x^*, y^*, dir^*, num^*)$ of the query $P(x^*) \to y^*$ or $P^{-1}(y^*) \to x^*$;
- An arbitrary record $(x^\circ, y^\circ, dir^\circ, num^\circ)$ that is created after $(x, y, dir, num)$.

Note that $num^\circ > num^* > num$ in this setting. Below we analyze the influences of $(x^*, y^*, dir^*, num^*)$ and $(x^\circ, y^\circ, dir^\circ, num^\circ)$ in two subsubsections respectively.

### 8.1.1 Influence of $(x^*, y^*, dir^*)$

First, (B-1) cannot be suddenly fulfilled after $(x^*, y^*, dir^*)$ is created: otherwise, there appears a bad 3-chain $\big((x_2, y_2), (x, y), (x_4, y_4)\big)$ with $(x, y) \in \Pi_{in}$, contradicting Lemma 2. Further, neither (B-4) nor (B-5) can be fulfilled, since they contradict **Inv1**. It thus remains to analyze (B-2) and (B-3). Depending on $dir$ and $dir^*$, there are four cases.

**Case 1: $dir \in \{\rightarrow, \perp_\rightarrow\}$, $dir^* =\rightarrow$.** This means $D$ queries $P(x^*) \rightarrow y^*$. We analyze the conditions w.r.t. $(x, y, dir)$ in turn.

*If (B-2) is fulfilled,* then $(x^*, y^*)$ and $(x, y)$ constitute a 3-chain $\big((x, y), (x_3, y_3), (x_4, y_4)\big)$. Now if $(x_3, y_3) = (x^*, y^*)$, it contradicts **Inv2** since $dir^* =\rightarrow$. If $(x_3, y_3) \neq (x^*, y^*)$ and $(x_4, y_4) = (x^*, y^*)$, i.e., $x^* = x_4 = y_3 \oplus \varphi(y \oplus x_3)$ then since $(x, y, dir) \in \Pi_{in}$ and $dir \in \{\rightarrow, \perp_\rightarrow\}$, $\mathcal{T}$ should have aborted at line 39 in $CheckPrivacy(x^*)$. Therefore, after $(x^*, y^*, \rightarrow)$ is created, (B-2) won't be suddenly fulfilled.

*If (B-3) is fulfilled* after $(x^*, y^*, \rightarrow, num^*)$ is created, then there appear two 2-chains $\big((x_2, y_2), (x, y)\big)$ and $\big((x, y), (x_3, y_3)\big)$ collide on either left or right. Further:

- Subcase 1.3.1: $(x_2, y_2) = (x_3, y_3) = (x^*, y^*)$. Then $x^* \oplus \varphi^{-1}(y^* \oplus x) = x \oplus \varphi^{-1}(y \oplus x^*)$ contradicts **Inv4**, whereas $y \oplus \varphi(y^* \oplus x) = y^* \oplus \varphi(y \oplus x^*)$ contradicts **Inv3**;

- Subcase 1.3.2: $(x_2, y_2) = (x^*, y^*)$ and $(x_3, y_3) \neq (x^*, y^*)$. Then, since $dir^* =\rightarrow$ and since we assumed $(x^*, y^*)$ the latest, $x^* \oplus \varphi^{-1}(y^* \oplus x) = x \oplus \varphi^{-1}(y \oplus x_3)$ again contradicts **Inv4**, whereas $y \oplus \varphi(y^* \oplus x) = y_3 \oplus \varphi(y \oplus x_3)$ contradicts **Inv3**;

- Subcase 1.3.3: $(x_3, y_3) = (x^*, y^*)$ and $(x_2, y_2) \neq (x^*, y^*)$. Then, $y \oplus \varphi(y_2 \oplus x) = y^* \oplus \varphi(y \oplus x^*)$ contradicts **Inv3**. On the other hand, if $x_2 \oplus \varphi^{-1}(y_2 \oplus x) = x \oplus \varphi^{-1}(y \oplus x^*)$, then since $(x, y, dir) \in \Pi_{in}$ and $dir \in \{\rightarrow, \perp_\rightarrow\}$, $\mathcal{T}$ should have aborted at line 41 in $CheckPrivacy(x^*)$.

Therefore, after $(x^*, y^*, \rightarrow)$ is created, (B-3) won't be suddenly fulfilled w.r.t. $(x, y)$.

**Case 2: $dir \in \{\rightarrow, \perp_\rightarrow\}$, $dir^* =\leftarrow$.** This means $D$ queries $P^{-1}(y^*) \rightarrow x^*$.

*If (B-2) is fulfilled,* then $(x^*, y^*)$, $(x, y)$ constitute a 3-chain $\big((x, y), (x_3, y_3), (x_4, y_4)\big)$. This always contradicts **Inv2**, regardless of $(x^*, y^*)$ equaling $(x_3, y_3)$ or $(x_4, y_4)$.

*If (B-3) is fulfilled* after $(x^*, y^*, \leftarrow, num^*)$ is created, then there appear two 2-chains $\big((x_2, y_2), (x, y)\big)$ and $\big((x, y), (x_3, y_3)\big)$ collide on either left or right. Further:

- Subcase 2.3.1: $(x_2, y_2) = (x_3, y_3) = (x^*, y^*)$. Then $x^* \oplus \varphi^{-1}(y^* \oplus x) = x \oplus \varphi^{-1}(y \oplus x^*)$ contradicts **Inv4**, whereas $y \oplus \varphi(y^* \oplus x) = y^* \oplus \varphi(y \oplus x^*)$ contradicts **Inv3**;

- Subcase 2.3.2: $(x_2, y_2) = (x^*, y^*)$, and $(x_3, y_3) \neq (x^*, y^*)$. Then $x^* \oplus \varphi^{-1}(y^* \oplus x) = x \oplus \varphi^{-1}(y \oplus x_3)$ contradicts **Inv4**, while $y \oplus \varphi(y^* \oplus x) = y_3 \oplus \varphi(y \oplus x_3)$ indicates $\mathcal{T}$ aborting at line 47 in $CheckPrivacy^{-1}(y^*)$;

- Subcase 2.3.3: $(x_3, y_3) = (x^*, y^*)$, and $(x_2, y_2) \neq (x^*, y^*)$. Then, since $dir^* =\leftarrow$, $y \oplus \varphi(y_2 \oplus x) = y^* \oplus \varphi(y \oplus x^*)$ contradicts **Inv3**, whereas $x_2 \oplus \varphi^{-1}(y_2 \oplus x) = x \oplus \varphi^{-1}(y \oplus x^*)$ contradicts **Inv4**.

Thus, after $(x^*, y^*, \leftarrow)$ is created, (B-3) won't be fulfilled w.r.t. $(x, y)$.

**Summary for $(x^*, y^*, dir^*)$.** Case 3, i.e., $dir \in \{\leftarrow, \perp_\leftarrow\}$, $dir^* =\rightarrow$, is similar to Case 2 by symmetry, while Case 4, i.e., $dir \in \{\leftarrow, \perp_\leftarrow\}$, $dir^* =\leftarrow$, is similar to Case 1 by symmetry. By the above, $(x, y)$ remains good after $(x^*, y^*)$ is created and added to $\Pi_{pub}$.

### 8.1.2 Influence of Arbitrary $(x^\circ, y^\circ, dir^\circ)$

For "internal" record $(x^\circ, y^\circ)$ added to $\Pi_{in}$ in this cycle, the analysis bears resemblance with the proof of Lemma 8. In detail,

- (B-1) cannot be fulfilled after $(x^\circ, y^\circ, dir^\circ, num^\circ)$ is created: otherwise, there appears a *bad* 3-chain $\big((x_2, y_2), (x, y), (x_4, y_4)\big)$ and it contradicts Lemma 2;
- (B-3) cannot be fulfilled, since $\mathcal{T}$ creating $(x^\circ, y^\circ)$ does not affect $\Pi_{pub}$ at all;
- Neither (B-4) nor (B-5) can be fulfilled, since they contradict **Inv1**.

It remains to address (B-2), and we distinguish two cases depending on *dir*.

**Case 1: $dir \in \{\rightarrow, \perp_\rightarrow\}$.** In this case, if (B-2) is fulfilled, then it means a 3-chain $\big((x, y), (x_3, y_3), (x_4, y_4)\big)$ suddenly appears after $(x^\circ, y^\circ, dir^\circ, num^\circ)$ is created. Further:
*Subcase 1.1: $(x_3, y_3) = (x^\circ, y^\circ)$.* $\big((x, y), (x^\circ, y^\circ), (x_4, y_4)\big)$ is bad and contradicts Lemma 2.
*Subcase 1.2: $(x_3, y_3) \in \Pi_{pub}$* (thus, $(x_3, y_3) \neq (x^\circ, y^\circ)$), $(x_4, y_4) = (x^\circ, y^\circ)$, and $dir^\circ \in \{\leftarrow, \perp_\leftarrow\}$. Then it contradicts **Inv2**.
*Subcase 1.3: $(x_3, y_3) \in \Pi_{pub}$, $(x_4, y_4) = (x^\circ, y^\circ)$, and $dir^\circ \in \{\rightarrow, \perp_\rightarrow\}$.* Then, right before $\mathcal{T}$ creating $(x^\circ, y^\circ)$, there exists $(x', y') \in \Pi_{pub}$ such that either of the following is fulfilled:

- The two 2-chains $\big((x', y'), (x^*, y^*)\big)$ and $\big((x, y), (x_3, y_3)\big)$ "collide at the right side", i.e., $y^* \oplus \varphi(y' \oplus x^*) = y_3 \oplus \varphi(y \oplus x_3)$. Since $(x_3, y_3), (x', y') \in \Pi_{pub}$, it holds $num^* \geq num_3, num'$; moreover, $num^* > num$ as remarked before. Therefore, it contradicts **Inv3**, regardless of $dir^* =\rightarrow$ or $\leftarrow$.
- The two 2-chains $\big((x^*, y^*), (x', y')\big)$ and $\big((x, y), (x_3, y_3)\big)$ "collide at the right side", i.e., $y' \oplus \varphi(y^* \oplus x') = y_3 \oplus \varphi(y \oplus x_3)$, and $(x', y') \neq (x^*, y^*)$. Then,
  - If $(x_3, y_3) = (x^*, y^*)$, then $y' \oplus \varphi(y^* \oplus x') = y^* \oplus \varphi(y \oplus x^*)$ contradicts **Inv3**;
  - If $(x_3, y_3) \neq (x^*, y^*)$ and $dir^* =\rightarrow$, it again contradicts **Inv3**;
  - If $(x_3, y_3) \neq (x^*, y^*)$ and $dir^* =\leftarrow$, then $y^* = x' \oplus y \oplus x_3 \oplus \varphi^{-1}(y' \oplus y_3)$. Since $dir \in \{\rightarrow, \perp_\rightarrow\}$, $\mathcal{T}$ should have aborted at line 47 in $CheckPrivacy^{-1}(y^*)$.

**Case 2: $dir \in \{\leftarrow, \perp_\leftarrow\}$.** In this case, if (B-2) is fulfilled, then it means a 3-chain $\big((x_1, y_1), (x_2, y_2), (x, y)\big)$ suddenly appears after $(x^\circ, y^\circ, dir^\circ, num^\circ)$ is created. The case-study is essentially similar to Case 1 by symmetry (except that some subcases contradict **Inv4** instead of **Inv3**). In all, after $(x^\circ, y^\circ)$ is created, (B-2) won't be fulfilled either. □

**Lemma 10.** *Assume that when $D$ queries $P(x^*) \to y^*$ ($P^{-1}(y^*) \to x^*$, resp.), it holds $x^* \in domain(\Pi_{in})$ ($y^* \in range(\Pi_{in})$, resp.), and there exists a record $(x, y, dir, num) \in \Pi_{in}$ that is good. Then, after the simulator cycle due to $D$ querying $P(x^*) \to y^*$ or $P^{-1}(y^*) \to x^*$, $(x, y, dir, num)$ remains good.*

*Proof.* For clarity, we list the query records that are relevant to the analysis:

- The record $(x, y, dir, num)$ created before $D$ querying $P(x^*) \to y^*$ or $P^{-1}(y^*) \to x^*$;
- The record $(x^*, y^*, dir^*, num^*) \in \Pi_{in}$ that corresponds to the adversarial query $P(x^*) \to y^*$ or $P^{-1}(y^*) \to x^*$ that triggers the current simulator cycle;
- An arbitrary record $(x^\circ, y^\circ, dir^\circ, num^\circ)$ that is created after $(x, y, dir, num)$.

In this setting, we have $num^\circ > num^*, num$, but it is unclear if $num^* > num$. Below we analyze the influence of $(x^*, y^*, dir^*, num^*)$ and $(x^\circ, y^\circ, dir^\circ, num^\circ)$ in turn.

### 8.1.3 Influence of $(x^*, y^*, dir^*)$

In this setting, the record $(x^*, y^*)$ is not new: it has been in either $\Pi_{in}$. By this, $\mathcal{T}$ moving $(x^*, y^*)$ to $\Pi_{pub}$ cannot make (B-1), (B-2), (B-4) or (B-5) fulfilled.

On the other hand, if (B-3) is fulfilled after $\mathcal{T}$ moving $(x^*, y^*)$ to $\Pi_{pub}$, then before $D$ queries $P(x^*)$ or $P^{-1}(y^*)$, there already existed two "colliding" 2-chains $\big((x_2, y_2), (x, y)\big)$ and $\big((x, y), (x_3, y_3)\big)$ such that either $(x_2, y_2) = (x^*, y^*)$ or $(x_3, y_3) = (x^*, y^*)$ (they cannot both hold: otherwise, $x^* \oplus \varphi^{-1}(y^* \oplus x) = x \oplus \varphi^{-1}(y \oplus x^*)$ contradicts **Inv4**, while $y \oplus \varphi(y^* \oplus x) = y^* \oplus \varphi(y \oplus x^*)$ contradicts **Inv3**). We distinguish four cases as follows.

**Case 1: $D$ queries $P(x^*)$, and $(x_2, y_2) = (x^*, y^*)$.** In this case, it has to be $dir^* \in \{\rightarrow, \perp_{\rightarrow}\}$: otherwise, $\mathcal{T}$ would have aborted at line 37 in $CheckPrivacy(x^*)$. But then, both the "left collision" $x^* \oplus \varphi^{-1}(y^* \oplus x) = x \oplus \varphi^{-1}(y \oplus x_3)$ and the "right collision" $y \oplus \varphi(y^* \oplus x) = y_3 \oplus \varphi(y \oplus x_3)$ would have caused $\mathcal{T}$ abort at line 52 in the call to $CheckInternalColl(x^*, y^*)$ (before it actually moved $(x^*, y^*)$ to $\Pi_{pub}$).

**Case 2: $D$ queries $P(x^*)$, and $(x_3, y_3) = (x^*, y^*)$.** In this case, it has to be $dir^* \in \{\rightarrow, \perp_{\rightarrow}\}$: otherwise, $\mathcal{T}$ would have aborted at line 37 in $CheckPrivacy(x^*)$. Then,

- $x_2 \oplus \varphi^{-1}(y_2 \oplus x) = x \oplus \varphi^{-1}(y \oplus x^*)$ means $\varphi^{-1}(x^*) = y_2 \oplus x \oplus y \oplus \varphi(x_2 \oplus x)$. Since $(x, y) \in \Pi_{in}$, $\mathcal{T}$ should have aborted at line 41 in $CheckPrivacy(x^*)$;
- $y \oplus \varphi(y_2 \oplus x) = y^* \oplus \varphi(y \oplus x^*)$ would have caused $\mathcal{T}$ abort at line 54 in the call to $CheckInternalColl(x^*, y^*)$.

**Case 3: $D$ queries $P^{-1}(y^*)$, and $(x_2, y_2) = (x^*, y^*)$.** This case is similar to Case 2 by symmetry. In detail, it has to be $dir^* \in \{\leftarrow, \perp_{\leftarrow}\}$: otherwise, $\mathcal{T}$ would have aborted at line 43 in $CheckPrivacy^{-1}(y^*)$. Then,

- $x^* \oplus \varphi^{-1}(y^* \oplus x) = x \oplus \varphi^{-1}(y \oplus x_3)$ would have caused $\mathcal{T}$ abort at line 61 in the call to $CheckInternalColl^{-1}(x^*, y^*)$, while
- $y \oplus \varphi(y^* \oplus x) = y_3 \oplus \varphi(y \oplus x_3)$ means $y^* = x \oplus y \oplus x_3 \oplus \varphi(y \oplus y_3)$. Since $(x, y) \in \Pi_{in}$, $\mathcal{T}$ should have aborted at line 47 in $CheckPrivacy^{-1}(y^*)$.

**Case 4: $D$ queries $P^{-1}(y^*)$, and $(x_3, y_3) = (x^*, y^*)$.** This case is similar to Case 1 by symmetry. A bit more clearly, it has to be $dir^* \in \{\leftarrow, \perp_{\leftarrow}\}$: otherwise, $\mathcal{T}$ would have aborted at line 43 in $CheckPrivacy^{-1}(y^*)$. But then, both the "left collision" $x_2 \oplus \varphi^{-1}(y_2 \oplus x) = x \oplus \varphi^{-1}(y \oplus x^*)$ and the "right collision" $y \oplus \varphi(y_2 \oplus x) = y^* \oplus \varphi(y \oplus x^*)$ would have caused $\mathcal{T}$ abort at line 59 in the call to $CheckInternalColl^{-1}(x^*, y^*)$.

**Summary for $(x^*, y^*, dir^*)$.** By the above, none of the conditions can be suddenly fulfilled after $\mathcal{T}$ moving $(x^*, y^*)$ to $\Pi_{pub}$, and $(x, y) \in \Pi_{in}$ remains good.

### 8.1.4 Influence of Arbitrary $(x^\circ, y^\circ, dir^\circ)$

The analysis for such $(x^\circ, y^\circ)$ bears some resemblance with the analogue part in the proof of Lemma 9. In detail,

- (B-1) cannot be fulfilled after $(x^\circ, y^\circ, dir^\circ, num^\circ)$ is created: otherwise, there appears a *bad* 3-chain $\big((x_2, y_2), (x, y), (x_4, y_4)\big)$ and it contradicts Lemma 2;
- (B-3) cannot be fulfilled, since $\mathcal{T}$ creating $(x^\circ, y^\circ)$ does not affect $\Pi_{pub}$ at all;
- Neither (B-4) nor (B-5) can be fulfilled, since they contradict **Inv1**.

It remains to consider (B-2), and we distinguish two cases depending on *dir*.

**Case 1: $D$ queries $P(x^*)$, and $dir \in \{\rightarrow, \perp_\rightarrow\}$.** In this case, it has to be $dir^* \in \{\rightarrow, \perp_\rightarrow\}$: otherwise, $\mathcal{T}$ would have aborted at line 37 in $CheckPrivacy(x^*)$.

Then, if (B-2) is fulfilled, it means a 3-chain $\big((x,y),(x_3,y_3),(x_4,y_4)\big)$ suddenly appears after $(x^\circ, y^\circ, dir^\circ, num^\circ)$ is created. We analyze each subcase as follows.

*Subcase 1.1: $(x_3,y_3) = (x^\circ,y^\circ)$.* Then since $(x^\circ,y^\circ) \in \Pi_{in}$, $\big((x,y),(x^\circ,y^\circ),(x_4,y_4)\big)$ is a bad 3-chain, and this contradicts Lemma 2;

*Subcase 1.2: $(x_3,y_3) \in \Pi_{pub}$, $(x_4,y_4) = (x^\circ,y^\circ)$, and $dir^\circ \in \{\leftarrow, \perp_\leftarrow\}$.* It contradicts **Inv2**.

*Subcase 1.3: $(x_3,y_3) \in \Pi_{pub}$, $(x_4,y_4) = (x^\circ,y^\circ)$, and $dir^\circ \in \{\rightarrow, \perp_\rightarrow\}$.* Then, right before $\mathcal{T}$ creating $(x^\circ,y^\circ)$, there exists $(x',y') \in \Pi_{pub}$ such that:

- The 2-chain $\big((x',y'),(x^*,y^*)\big)$ has its "right endpoint" collides with the 2-chain $\big((x,y),(x_3,y_3)\big)$, i.e., $y^* \oplus \varphi(y' \oplus x^*) = y_3 \oplus \varphi(y \oplus x_3)$; or
- The 2-chain $\big((x^*,y^*),(x',y')\big)$ has its "right endpoint" collides with the 2-chain $\big((x,y),(x_3,y_3)\big)$, i.e., $y' \oplus \varphi(y^* \oplus x') = y_3 \oplus \varphi(y \oplus x_3)$.

Though, since $dir^* \in \{\rightarrow, \perp_\rightarrow\}$, both possibilities would have caused $\mathcal{T}$ abort (at line 54 or 52) in the call to $CheckInternalColl(x^*,y^*)$.

**Case 2: $D$ queries $P^{-1}(y^*)$, and $dir \in \{\rightarrow, \perp_\rightarrow\}$.** In this case, it has to be $dir^* \in \{\leftarrow, \perp_\leftarrow\}$: otherwise, $\mathcal{T}$ would have aborted at line 43 in $CheckPrivacy^{-1}(y^*)$.

Then, if (B-2) is fulfilled, it means a 3-chain $\big((x,y),(x_3,y_3),(x_4,y_4)\big)$ suddenly appears after $(x^\circ, y^\circ, dir^\circ, num^\circ)$ is created. We analyze each subcase as follows.

*Subcase 2.1: $(x_3,y_3) = (x^\circ,y^\circ)$.* Then since $(x^\circ,y^\circ) \in \Pi_{in}$, $\big((x,y),(x^\circ,y^\circ),(x_4,y_4)\big)$ is a bad 3-chain, and this contradicts Lemma 2.

*Subcase 2.2: $(x_3,y_3) \in \Pi_{pub}$, $(x_4,y_4) = (x^\circ,y^\circ)$, and $dir^\circ \in \{\leftarrow, \perp_\leftarrow\}$.* It contradicts **Inv2**.

*Subcase 2.3: $(x_3,y_3) \in \Pi_{pub}$, $(x_4,y_4) = (x^\circ,y^\circ)$, and $dir^\circ \in \{\rightarrow, \perp_\rightarrow\}$.* Then, right before $\mathcal{T}$ creating $(x^\circ,y^\circ)$, there exists $(x',y') \in \Pi_{pub}$ such that:

- The 2-chain $\big((x',y'),(x^*,y^*)\big)$ has its "right endpoint" collides with the 2-chain $\big((x,y),(x_3,y_3)\big)$, i.e., $y^* \oplus \varphi(y' \oplus x^*) = y_3 \oplus \varphi(y \oplus x_3)$. Then, since $dir^* \in \{\leftarrow, \perp_\leftarrow\}$, $\mathcal{T}$ would have aborted at line 59 in the call $CheckInternalColl^{-1}(x^*,y^*)$.
- Or: the 2-chain $\big((x^*,y^*),(x',y')\big)$ has its "right endpoint" collides with the 2-chain $\big((x,y),(x_3,y_3)\big)$, i.e., $y' \oplus \varphi(y^* \oplus x') = y_3 \oplus \varphi(y \oplus x_3)$. Then, $\mathcal{T}$ would have aborted at line 47 in $CheckPrivacy^{-1}(y^*)$.

**Case 3 and 4, and Summary.** Case 3, where $D$ queries $P(x^*)$, and $dir \in \{\leftarrow, \perp_\leftarrow\}$, is essentially similar to Case 2 by symmetry. Case 4, where $D$ queries $P^{-1}(y^*)$, and $dir \in \{\leftarrow, \perp_\leftarrow\}$, is similar to Case 1 by symmetry. By the above, in this cycle, no newly created record can make $(x,y)$ bad. □

We conclude with the following lemma.

**Lemma 11.** *Right before every call to $ProcessRecord(x,y,dir)$, the record $(x,y)$ is good.*

*Proof.* By Lemma 7, $(x,y)$ was good right after it was created. If $(x,y)$ was created in a new cycle, then $ProcessRecord(x,y,dir)$ is called immediately after $(x,y)$ was created, and the claim thus holds. Otherwise, Lemmas 8, 9 and 10 imply that $(x,y) \in \Pi_{in}$ remains good after subsequent simulator actions, till the corresponding transferring cycle. Therefore, $(x,y)$ remains good before the call to $ProcessRecord(x,y,dir)$. □

## 8.2 Abort Probability of Adaptations

**Lemma 12.** *Consider a call to $ProcessRecord(x, y, dir)$. If the record $(x, y, dir, num)$ is good right before this call, then in each subsequent call to Adapt, the probability that $\mathcal{T}$ aborts is at most $3q^2/(2^n - q^2)$.*

*Proof.* Wlog, consider $dir \in \{\rightarrow, \perp_\rightarrow\}$. See Fig. 2 for a summary of the 2-chains and 3-chains considered by $ProcessRecord(x, y, dir)$. Since $(x, y, dir, num)$ is good,

- For every 2-chain $\big((x_2^{(i)}, y_2^{(i)}), (x, y)\big)$, we have $x_4^{(i)} = y \oplus \varphi(y_2^{(i)} \oplus x) \notin domain(\Pi_{all})$ by $\neg$(B-1); (Though, it might hold $(x_1^{(i)}, y_1^{(i)}) \in \Pi_{all}$ for $y_1^{(i)} = x_2^{(i)} \oplus \varphi^{-1}(y_2^{(i)} \oplus x)$.)
- For every 2-chain $\big((x, y), (x_3^{(i)}, y_3^{(i)})\big)$, we have $y_1^{(i)} = x \oplus \varphi^{-1}(y \oplus x_3^{(i)}) \notin range(\Pi_{all})$ by $\neg$(B-1) and $x_4^{(i)} = y_3^{(i)} \oplus \varphi(y \oplus x_3^{(i)}) \notin domain(\Pi_{all})$ by $\neg$(B-2).

By these,

- In $Check3Chains(x, y, dir)$, $\mathcal{T}$ only detects 3-chains $\big((x_1^{(i)}, y_1^{(i)}), (x_2^{(i)}, y_2^{(i)}), (x, y)\big)$;
- In $Check2Chains(x, y)$, the first forall $(x', y') \in \Pi_{pub}$ loop only detects 2-chains $\big((x_2^{(i)}, y_2^{(i)}), (x, y)\big)$ with $y_1^{(i)} = x_2^{(i)} \oplus \varphi^{-1}(y_2^{(i)} \oplus x) \notin range(\Pi_{all})$ (otherwise, $\mathcal{T}$ would have added $\big((x_2^{(i)}, y_2^{(i)}), (x, y)\big)$ to $CompletedChains$ in $Check3Chains(x, y, dir)$); the second forall $(x', y') \in \Pi_{pub}$ loop only detects 2-chains $\big((x, y), (x_3^{(i)}, y_3^{(i)})\big)$.

We proceed to bound abort probabilities of adaptations in each of the subsequent steps.

**Adaptations during completing 3-chains.** For $i = 1, ..., \alpha$, $\mathcal{T}$ considers the $i$-th 3-chain $\big((x_1^{(i)}, y_1^{(i)}), (x_2^{(i)}, y_2^{(i)}), (x, y)\big)$, and computes $k^{(i)} \leftarrow \varphi^{-1}(y_1^{(i)} \oplus x_2^{(i)})$, $x_4^{(i)} \leftarrow y \oplus \varphi^3(k^{(i)})$, $u^{(i)} \leftarrow k^{(i)} \oplus x_1^{(i)}$, $v^{(i)} \leftarrow E\big(k^{(i)}, u^{(i)}\big)$ and $y_4^{(i)} \leftarrow \varphi^4(k^{(i)}) \oplus v^{(i)}$. $\mathcal{T}$ finally makes a call to $Adapt(x_4^{(i)}, y_4^{(i)}, \perp_\rightarrow, num^{(i)})$, yielding 4-chain $\big((x_1^{(i)}, y_1^{(i)}), (x_2^{(i)}, y_2^{(i)}), (x, y), (x_4^{(i)}, y_4^{(i)})\big)$. The $Adapt$-call aborts if $x_4^{(i)} \in domain(\Pi_{all})$ or $y_4^{(i)} \in range(\Pi_{all})$.

Before the call to $ProcessRecord(x, y, dir)$, it holds $x_4^{(i)} \notin domain(\Pi_{all})$ by $\neg$(B-1). In addition, it clearly holds $x_4^{(i)} \neq x_4^{(i')}$ for any $i' \in \{1, ..., i-1\}$ $\big($since $x_4^{(i)} = y \oplus \varphi(y_2^{(i)} \oplus x)$ and $x_4^{(i')} = y \oplus \varphi(y_2^{(i')} \oplus x)$, and since $y_2^{(i)} \neq y_2^{(i')}\big)$, and the (adapted) records $(x_4^{(i')}, y_4^{(i')})$ created earlier in this $ProcessRecord$-call won't add $x_4^{(i)}$ to $domain(\Pi_{all})$. Therefore, $x_4^{(i)} \notin domain(\Pi_{all})$ right before the call to $Adapt(x_4^{(i)}, y_4^{(i)}, \perp_\rightarrow, num^{(i)})$.

On the other hand, by Lemma 5, the query $E\big(k^{(i)}, u^{(i)}\big) \rightarrow v^{(i)}$ is new. By this and by $|ET| \le q^2$, $\Pr[y_4^{(i)} \in range(\Pi_{all})] \le |\Pi_{all}|/(2^n - q^2) \le 3q^2/(2^n - q^2)$. In summary, $Adapt(x_4^{(i)}, y_4^{(i)}, \perp_\rightarrow, num^{(i)})$ aborts at line 119 with probability $\le 3q^2/(2^n - q^2)$.

**Adaptations during completing 2-chains $\big((x_2^{(\alpha+i)}, y_2^{(\alpha+i)}), (x, y)\big)$.** For $i = 1, ..., \beta$, $\mathcal{T}$ considers the $i$-th 2-chain $\big((x_2^{(\alpha+i)}, y_2^{(\alpha+i)}), (x, y)\big)$, computes $k^{(\alpha+i)} \leftarrow \varphi^{-2}(y_2^{(\alpha+i)} \oplus x)$, $y_1^{(\alpha+i)} \leftarrow \varphi^1(k^{(\alpha+i)}) \oplus x_2^{(\alpha+i)}$, $x_4^{(\alpha+i)} \leftarrow y \oplus \varphi^3(k^{(\alpha+i)})$, $y_4^{(\alpha+i)} \leftarrow InP(x_4^{(\alpha+i)})$, $v^{(\alpha+i)} \leftarrow \varphi^4(k^{(\alpha+i)}) \oplus y_4^{(\alpha+i)}$, $u^{(\alpha+i)} \leftarrow E^{-1}\big(k^{(\alpha+i)}, v^{(\alpha+i)}\big)$ and $x_1^{(\alpha+i)} \leftarrow k^{(\alpha+i)} \oplus u^{(\alpha+i)}$. $\mathcal{T}$ finally makes a call to $Adapt(x_1^{(\alpha+i)}, y_1^{(\alpha+i)}, \perp_\leftarrow, num^{(\alpha+i)})$ to complete the 4-chain $\big((x_1^{(\alpha+i)}, y_1^{(\alpha+i)}, \perp_\leftarrow), (x_2^{(\alpha+i)}, y_2^{(\alpha+i)}), (x, y), (x_4^{(\alpha+i)}, y_4^{(\alpha+i)})\big)$.

We focus on $Adapt(x_1^{(\alpha+i)}, y_1^{(\alpha+i)}, \perp_\leftarrow, num^{(\alpha+i)})$. At the right side, for every $i$, it holds $y_1^{(\alpha+i)} \in range(\Pi_{all})$ before this simulator cycle: otherwise, the records $(x_2^{(\alpha+i)}, y_2^{(\alpha+i)})$ and $(x, y)$ would have been in a 3-chain rather than the 2-chain. On the other hand,

- $y_1^{(\alpha+i)} \neq y_1^{(i')}$ for all $i' \in \{1, ..., \alpha + i - 1\}$ by ¬(B-4). Therefore, each record $(x_1^{(i')}, y_1^{(i')}, dir^{(i')})$ with $dir^{(i')} \in \{\leftarrow, \perp_\leftarrow\}$ created earlier in this *ProcessRecord*-call won't add $y_1^{(\alpha+i)}$ to $range(\Pi_{all})$; and
- Each record $(x', y', dir')$ with $dir' \in \{\rightarrow, \perp_\rightarrow\}$ created earlier in this *ProcessRecord*-call has $y' \neq y_1^{(\alpha+i)}$ by **Inv2**, and won't add $y_1^{(\alpha+i)}$ to $range(\Pi_{all})$ either.

By these, it remains $y_1^{(\alpha+i)} \in range(\Pi_{all})$ till $Adapt(x_1^{(\alpha+i)}, y_1^{(\alpha+i)}, \perp_\leftarrow, num^{(\alpha+i)})$.

On the other hand, by Lemma 5, the query $E^{-1}(k^{(\alpha+i)}, v^{(\alpha+i)}) \rightarrow u^{(\alpha+i)}$ is new, and $\Pr[x_1^{(\alpha+i)} \in domain(\Pi_{all})] \leq 3q^2/(2^n - q^2)$. In all, $Adapt(x_1^{(\alpha+i)}, y_1^{(\alpha+i)}, \perp_\leftarrow, num^{(\alpha+i)})$ aborts at line 119 with probability $\leq 3q^2/(2^n - q^2)$.

**Adaptations during completing 2-chains $\big((x, y), (x_3^{(\alpha+\beta+i)}, y_3^{(\alpha+\beta+i)})\big)$.** For $i = 1, ..., \gamma$, $j = \alpha + \beta + i$, $\mathcal{T}$ considers the $i$-th 2-chain $\big((x, y), (x_3^{(j)}, y_3^{(j)})\big)$, and computes $k^{(j)} \leftarrow \varphi^{-2}(y \oplus x_3^{(j)})$, $y_1^{(j)} \leftarrow \varphi^1(k^{(j)}) \oplus x$, $x_1^{(j)} \leftarrow InP^{-1}(y_1^{(j)})$, $u^{(j)} \leftarrow k^{(j)} \oplus x_1^{(j)}$, $v^{(j)} \leftarrow E(k^{(j)}, u^{(j)})$, $y_4^{(j)} \leftarrow \varphi^4(k^{(j)}) \oplus v^{(j)}$ and $x_4^{(j)} \leftarrow \varphi^3(k^{(j)}) \oplus y_3^{(j)}$. The simulator $\mathcal{T}$ finally makes a call to $Adapt(x_4^{(j)}, y_4^{(j)}, \perp_\rightarrow, num^{(j)})$.

We focus on the call $Adapt(x_4^{(j)}, y_4^{(j)}, \perp_\rightarrow, num^{(j)})$. At the left side, for every $i$, it holds $x_4^{(j)} \in domain(\Pi_{all})$ before this cycle by ¬(B-2). At the right,

- $x_4^{(j)} \neq x_4^{(i')}$ for all $i' \in \{1, ..., \alpha + \beta + i - 1\}$ by ¬(B-3) and ¬(B-5). Therefore, each record $(x_4^{(i')}, y_4^{(i')}, dir^{(i')})$ with $dir^{(i')} \in \{\rightarrow, \perp_\rightarrow\}$ created earlier in this *ProcessRecord*-call won't add $x_4^{(j)}$ to $domain(\Pi_{all})$; and
- Each record $(x', y', dir')$ with $dir' \in \{\leftarrow, \perp_\leftarrow\}$ created earlier in this *ProcessRecord*-call has $x' \neq x_4^{(j)}$ by **Inv2**, and won't add $x_4^{(j)}$ to $domain(\Pi_{all})$ either.

By these, $x_4^{(j)} \in domain(\Pi_{all})$ till $Adapt(x_4^{(j)}, y_4^{(j)}, \perp_\rightarrow, num^{(j)})$.

Once again, by Lemma 5, the query $E\big(k^{(j)}, u^{(j)}\big) \rightarrow v^{(j)}$ is new, and $\Pr[y_4^{(j)} \in range(\Pi_{all})] \leq 3q^2/(2^n - q^2)$. $Adapt(x_4^{(j)}, y_4^{(j)}, \perp_\rightarrow, num^{(j)})$ thus aborts with probability at most $3q^2/(2^n - q^2)$. Thus the claim. □

We thereby obtain the abort probability due to adaptations.

**Lemma 13.** *The probability that $\mathcal{T}$ aborts inside the procedure Adapt is at most $\frac{3q^4}{2^n - q^2}$.*

*Proof.* By Lemma 11, $\mathcal{T}$ always calls $ProcessRecord(x, y, dir)$ with good $(x, y)$. Therefore, during the $\Sigma_2$ execution, every *Adapt*-call aborts with probability $\leq 3q^2/(2^n - q^2)$ by Lemma 12. By Lemma 3, the number of *Adapt*-calls is $\leq q^2$. Thus the claim. □

## 8.3 Summary on Abort Probability

**Lemma 14.** *The probability that $\Sigma_2$ aborts in procedures $InP/InP^{-1}$ is at most $4q^4/2^n$.*

*Proof.* Inside a call to $InP(x)$, $\mathcal{T}$ may query $\mathbf{p}(x) \rightarrow y$ and adds the record $(x, y, \rightarrow, qnum)$ to $\Pi_{in}$. At this time, $\mathcal{T}$ aborts if there already existed $(x', y') \in \Pi_{all}$ such that $y' = y$. Such $(x', y')$ must be adapted: otherwise $y' = y$ is impossible. By Lemma 3, the number of adapted records is $\leq q^2$, while the number of non-adapted is at most $q^2 + q \leq 2q^2$. Therefore, a call to $InP(x)$ aborts with probability $\leq q^2/(2^n - 2q^2)$. Similarly, a call to $InP^{-1}(y)$ aborts with probability $\leq q^2/(2^n - 2q^2)$. Thus, assuming $2q^2 \leq 2^n/2$, we have

$$\Pr\big[InP \text{ and } InP^{-1} \text{ abort}\big] \leq (q + q^2) \cdot \frac{q^2}{2^n - 2q^2} \leq \frac{2q^4}{2^n - 2q^2} \leq \frac{4q^4}{2^n} \tag{6}$$

since $InP(x)/InP^{-1}(y)$ are called at most $q^2+q$ times by Lemma 3. □

In Sect. 8.1, we have shown that $ProcessRecord(x, y, dir)$ is only called with good $(x, y)$ as long as "bad events" never occur. Therefore, summing over the probability of bad events of Lemmas 4, 6, 13 and 14 and of Lemma 12, we are able to conclude on the abort probability of $\mathcal{T}^{E,\mathbf{p}}$ in the intermediate system $\Sigma_2$.

**Lemma 15.** *The probability that $\mathcal{T}^{E,\mathbf{p}}$ aborts in $D^{\Sigma_2}$ is at most $\big(762C(\varphi)q^7+1362q^{10}\big)/2^n$.*

# 9 Indistinguishability of $\Sigma_1$ and $\Sigma_3$

**$\Sigma_1$ to $\Sigma_2$.** We first prove a helper lemma establishing the soundness of the simulation.

**Lemma 16.** *Consider an arbitrary blockcipher query $EMSP[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}}(k,u) \to v$ (resp., $(EMSP[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}})^{-1}(k,v) \to u$) that is made by $D$ in a $\Sigma_2$ execution. Then, when $D$ has received the answer, there exists a 4-chain $\big((x_1,y_1),(x_2,y_2),(x_3,y_3),(x_4,y_4)\big) \in (\Pi_{all})^4$ such that $E(k,u)=v$, $x_1=k\oplus u$ and $y_4=\varphi^4(k)\oplus v$ correspondingly.*

*Proof.* Wlog consider $D$ querying $\mathrm{EMSP}[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}}(k,u) \to v$ (backward queries are similar).

*Case 1: $(k,u,v) \in ET$ when $D$ queries $EMSP[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}}(k,u)$.* Then, $(k,u,v) \in ET$ is necessarily due to the simulator $\mathcal{T}^{E,\mathbf{p}}$ querying $E(k,u)$ or $E^{-1}(k,v)$ at some earlier time. Wlog assume that $\mathcal{T}^{E,\mathbf{p}}$ queried the forward $E(k,u)$. By construction, this only happens in a call to $Complete^+(y_3,k)$. When this call returns without abortion, there exists a 4-chain $\big((x_1,y_1),(x_2,y_2),(x_3,y_3),(x_4,y_4)\big) \in (\Pi_{all})^4$ such that $E(k,u)=v$, $x_1=k\oplus u$ and $y_4=\varphi^4(k)\oplus v$. Therefore, when $D$ queries $\mathrm{EMSP}[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}}(k,u)$, the 4-chain exists.

*Case 2: $(k,u,v) \notin ET$ when $D$ queries $EMSP[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}}(k,u)$.* To evaluate $\mathrm{EMSP}[\varphi]_4^{\mathcal{T}}(k,u)$, $\mathrm{EMSP}[\varphi]_4$ proceeds with $x_1 \leftarrow k\oplus u$, $\mathcal{T}^{E,\mathbf{p}}.P(x_1)\to y_1$, $x_2 \leftarrow \varphi(k)\oplus y_1$, $\mathcal{T}^{E,\mathbf{p}}.P(x_2)\to y_2$, $x_3 \leftarrow \varphi^2(k)\oplus y_2$, $\mathcal{T}^{E,\mathbf{p}}.P(x_3)\to y_3$, $x_4 \leftarrow \varphi^3(k)\oplus y_3$, $\mathcal{T}^{E,\mathbf{p}}.P(x_4)\to y_4$ and finally $v \leftarrow \varphi^4(k)\oplus y_4$. After $\mathrm{EMSP}[\varphi]_4$ receives the response $y_3$ for its third query to $\mathcal{T}^{E,\mathbf{p}}$, it holds $(x_2,y_2),(x_3,y_3) \in \Pi_{pub}$. By Lemma 1, the 2-chain $\big((x_2,y_2),(x_3,y_3)\big)$ has been in a 4-chain $\big((x_1,y_1),(x_2,y_2),(x_3,y_3),(x_4,y_4)\big)$ with $E(k,k\oplus x_1)=\varphi^4(k)\oplus y_4$. □

With Lemma 16, we are able to establish indistinguishability of $\Sigma_1$ and $\Sigma_2$.

**Lemma 17.** *For any distinguisher $D$ of total oracle query cost $q$, it holds*

$$\Big|\Pr\big[D^{\Sigma_1(E,\mathcal{S}^{E,\mathbf{p}})}=1\big]-\Pr\big[D^{\Sigma_2(EMSP[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}},\mathcal{T}^{E,\mathbf{p}})}=1\big]\Big| \le \frac{762C(\varphi)q^7+1362q^{10}}{2^n}.$$

*Proof.* In $\Sigma_1$ and $\Sigma_2$, the sequential distinguisher $D$ necessarily first queries $\mathcal{S}^{E,\mathbf{p}}$ (in $\Sigma_1$) or $\mathcal{T}^{E,\mathbf{p}}$ (in $\Sigma_2$) and then $E$ (in $\Sigma_1$) or $\mathrm{EMSP}[\varphi]_4$ (in $\Sigma_2$) only. Thus, the transcript of the first phase of the interaction (i.e., queries of $D$ to $\mathcal{S}^{E,\mathbf{p}}$) are clearly the same, since in both cases they are answered by $\mathcal{S}^{E,\mathbf{p}}$ and $\mathcal{T}^{E,\mathbf{p}}$ using the same randomness $(E,\mathbf{p})$ and essentially the same actions. In the second phase where $D$ queries its left oracle, Lemma 16 ensures that for every forward query $(k,u)$ (resp., backward query $(k,v)$), $D$ receives identical responses $v=E(k,u)=\mathrm{EMSP}[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}}(k,u)$ (resp., $u=E^{-1}(k,v)=(\mathrm{EMSP}[\varphi]_4^{-1})^{\mathcal{T}^{E,\mathbf{p}}}(k,v)$) in both $\Sigma_1$ and $\Sigma_2$ executions. Hence, the transcripts of the interaction of $D$ with $\Sigma_1(E,\mathbf{p})$ and $\Sigma_2(E,\mathbf{p})$ are the same for any good tuple $(E,\mathbf{p})$. Further using Lemma 15 yields $\big|\Pr[D^{\Sigma_1}=1]-\Pr[D^{\Sigma_2}=1]\big| \le \Pr[(E,\mathbf{p})\ \textit{is bad}] \le \frac{762C(\varphi)q^7+1362q^{10}}{2^n}$ as claimed. □

**$\boldsymbol{\Sigma_2}$ to $\boldsymbol{\Sigma_3}$.** We follow [CS15] and define a map $\Lambda$ mapping pairs $(E, \mathbf{p})$ either to the special symbol $\perp$ when $(E, \mathbf{p})$ is bad, or to a *partial permutation* $\mathbf{p}'$ when $(E, \mathbf{p})$ is good. A partial permutation is functions $\mathbf{p}'\colon \{0,1\}^n \to \{0,1\}^n \cup \{*\}$ and $\mathbf{p}'^{-1}\colon \{0,1\}^n \to \{0,1\}^n \cup \{*\}$, such that for all $x, y \in \{0,1\}^n$, $\mathbf{p}'(x) = y \neq * \Leftrightarrow \mathbf{p}'^{-1}(y) = x \neq *$.

Then map $\Lambda$ is defined for good pairs $(E, \mathbf{p})$ as follows: run $D^{\Sigma_2(E,\mathbf{p})}$, and consider the set $\Pi_{all}$ of the simulator at the end of the execution: then fill all undefined entries of the $\Pi_{all}$'s with the special symbol $*$. The result is exactly $\Lambda(E, \mathbf{p})$. By design of our simulators, the set $\Pi_{all}$ and thus $\Lambda(E, \mathbf{p})$ is a partial permutation as just defined above. We say that a partial permutation $\mathbf{p}'$ is good if it has a good preimage by $\Lambda$. Then, we say that a permutation $\mathbf{p}$ extends a partial permutation $\mathbf{p}'$, denoted $\mathbf{p} \vdash \mathbf{p}'$, if $\mathbf{p}$ and $\mathbf{p}'$ agree on all entries such that $\mathbf{p}'(x) \neq *$ and $\mathbf{p}'^{-1}(y) \neq *$.

By definition of $\Lambda$, for any good tuple of partial permutations $\mathbf{p}'$, the outputs of $D^{\Sigma_2(E,\mathbf{p})}$ and $D^{\Sigma_3(\mathbf{p})}$ are equal for any pair $(E, \mathbf{p})$ such that $\Lambda(E, \mathbf{p}) = \mathbf{p}'$ and any permutations $\mathbf{p}$ such that $\mathbf{p} \vdash \mathbf{p}'$. Let $\Omega_2$ be the set of partial permutations $\mathbf{p}'$ such that $D^{\Sigma_2(E,\mathbf{p})}$ output 1 for any $(E, \mathbf{p})$ with $\Lambda(E, \mathbf{p}) = \mathbf{p}'$. Then, we have the following ratio.

**Lemma 18.** *For any distinguisher $D$ of total oracle query cost $q$ and any $\mathbf{p}' \in \Omega_2$, it holds*

$$\frac{\Pr\big[\mathbf{p} \vdash \mathbf{p}'\big]}{\Pr\big[\Lambda(E, \mathbf{p}) = \mathbf{p}'\big]} \geq 1 - \frac{q^4}{2^n}.$$

*Proof.* First, since the number of "non-empty" entries $\mathbf{p}'(x) = y \neq *$ is $|\Pi_{all}|$, we have $\Pr\big[\mathbf{p} \vdash \mathbf{p}'\big] = \prod_{j=0}^{|\Pi_{all}|-1} \frac{1}{2^n - j}$. For the rest, fix any good preimage $(\widetilde{E}, \widetilde{\mathbf{p}})$ of $\widetilde{\mathbf{p}'}$. One can check that for any tuple $(E, \mathbf{p})$, $\Lambda(E, \mathbf{p}) = \mathbf{p}'$ *iff* the transcript of the interaction of $\mathcal{T}$ with $(E, \mathbf{p})$ in $D^{\Sigma_2(E,\mathbf{p})}$ is the same as the transcript of the interaction of $\mathcal{T}$ with $(\widetilde{E}, \widetilde{\mathbf{p}})$ in $D^{\Sigma_2(\widetilde{E},\widetilde{\mathbf{p}})}$. Assume that during the $\Sigma_2$ execution $D^{\Sigma_2(\mathrm{EMSP}[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}}, \mathcal{T}^{E,\mathbf{p}})}$, $\mathcal{T}$ makes $q_e$ and $q_1$ queries to $E$ and $\mathbf{p}$ respectively. Then,

$$\Pr\big[\Lambda(E, \mathbf{p}) = \mathbf{p}'\big] \leq \Big(\prod_{j=0}^{q_e-1} \frac{1}{2^n - j}\Big)\Big(\prod_{j=0}^{q_1-1} \frac{1}{2^n - j}\Big).$$

By the design of $\mathcal{T}$, it is easy to see that $q_e + q_1 = |\Pi_{all}|$: because $q_1$ equal the number of lazily sampled records in $\Pi_{all}$, while $q_e$ equal the number of adapted records in $\Pi_{in}$. Furthermore, using $|ET| \leq q^2$ yields

$$\frac{\Pr\big[\mathbf{p} \vdash \mathbf{p}'\big]}{\Pr\big[\Lambda(E, \mathbf{p}) = \mathbf{p}'\big]} \geq \frac{\prod_{j=0}^{|\Pi_{all}|-1} \frac{1}{2^n - j}}{\big(\prod_{j=0}^{q_e-1} \frac{1}{2^n - j}\big)\big(\prod_{j=0}^{q_1-1} \frac{1}{2^n - j}\big)} \geq \prod_{j=0}^{q^2-1} \Big(1 - \frac{j}{2^n}\Big) \geq 1 - \frac{q^4}{2^n}$$

as claimed. □

**Lemma 19.** *For any distinguisher $D$ with total oracle query cost at most $q$, it holds*

$$\Big|\Pr\big[D^{\Sigma_2(EMSP[\varphi]_4^{\mathcal{T}^{E,\mathbf{p}}}, \mathcal{T}^{E,\mathbf{p}})} = 1\big] - \Pr\big[D^{\Sigma_3(EMSP[\varphi]_4^{\mathbf{p}}, \mathbf{p})} = 1\big]\Big| \leq \frac{762C(\varphi)q^7 + 1363q^{10}}{2^n}.$$

*Proof.* Gathering Lemmas 15 and 18, the left hand side is bounded by

$$\begin{aligned}
&\leq \Pr\big[(E, \mathbf{p})\ \textit{is bad}\big] + \sum_{\mathbf{p}' \in \Omega_2} \Pr\big[\Lambda(E, \mathbf{p}) = \mathbf{p}'\big] - \sum_{\mathbf{p}' \in \Omega_2} \Pr\big[\mathbf{p} \vdash \mathbf{p}'\big] \\
&\leq \Pr\big[(E, \mathbf{p})\ \textit{is bad}\big] + \sum_{\mathbf{p}' \in \Omega_2} \Pr\big[\Lambda(E, \mathbf{p}) = \mathbf{p}'\big] \bigg(1 - \frac{\Pr\big[\mathbf{p} \vdash \mathbf{p}'\big]}{\Pr\big[\Lambda(E, \mathbf{p}) = \mathbf{p}'\big]}\bigg) \\
&\leq \frac{762C(\varphi)q^7 + 1362q^{10}}{2^n} + \frac{q^4}{2^n} \leq \frac{762C(\varphi)q^7 + 1363q^{10}}{2^n}
\end{aligned}$$

as claimed. □

Gathering Lemmas 3, 17 and 19 yields the bound in Theorem 1.

## Acknowledgments

We sincerely thank the anonymous reviewers for their helpful comments. Chun Guo was partly supported by the National Natural Science Foundation of China (Grant No. 62002202) and the Taishan Scholars Program (for Young Scientists) of Shandong.

## References

[ABD$^+$13] Elena Andreeva, Andrey Bogdanov, Yevgeniy Dodis, Bart Mennink, and John P. Steinberger. On the indifferentiability of key-alternating ciphers. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 531–550. Springer, Heidelberg, August 2013.

[ABM14] Elena Andreeva, Andrey Bogdanov, and Bart Mennink. Towards understanding the known-key security of block ciphers. In Shiho Moriai, editor, *FSE 2013*, volume 8424 of *LNCS*, pages 348–366. Springer, Heidelberg, March 2014.

[BKL$^+$12] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, François-Xavier Standaert, John P. Steinberger, and Elmar Tischhauser. Key-alternating ciphers in a provable setting: Encryption using a small number of public permutations - (extended abstract). In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 45–62. Springer, Heidelberg, April 2012.

[BKN09] Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic. Distinguisher and related-key attack on the full AES-256. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 231–249. Springer, Heidelberg, August 2009.

[Bla06] John Black. The ideal-cipher model, revisited: An uninstantiable blockcipher-based hash function. In Matthew J. B. Robshaw, editor, *FSE 2006*, volume 4047 of *LNCS*, pages 328–340. Springer, Heidelberg, March 2006.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.

[CHK$^+$16] Jean-Sébastien Coron, Thomas Holenstein, Robin Künzler, Jacques Patarin, Yannick Seurin, and Stefano Tessaro. How to build an ideal cipher: The indifferentiability of the Feistel construction. *Journal of Cryptology*, 29(1):61–114, January 2016.

[CLL$^+$18] Shan Chen, Rodolphe Lampe, Jooyoung Lee, Yannick Seurin, and John P. Steinberger. Minimizing the two-round Even-Mansour cipher. *Journal of Cryptology*, 31(4):1064–1119, October 2018.

[CS14] Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 327–350. Springer, Heidelberg, May 2014.

[CS15] Benoit Cogliati and Yannick Seurin. On the provable security of the iterated Even-Mansour cipher against related-key and chosen-key attacks. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 584–613. Springer, Heidelberg, April 2015.

[CS16] Benoît Cogliati and Yannick Seurin. Strengthening the known-key security notion for block ciphers. In Thomas Peyrin, editor, *FSE 2016*, volume 9783 of *LNCS*, pages 494–513. Springer, Heidelberg, March 2016.

[DKS12] Orr Dunkelman, Nathan Keller, and Adi Shamir. Minimalism in cryptography: The Even-Mansour scheme revisited. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 336–354. Springer, Heidelberg, April 2012.

[DRST12] Yevgeniy Dodis, Thomas Ristenpart, John P. Steinberger, and Stefano Tessaro. To hash or not to hash again? (In)differentiability results for $H^2$ and HMAC. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 348–366. Springer, Heidelberg, August 2012.

[DSST17] Yuanxi Dai, Yannick Seurin, John P. Steinberger, and Aishwarya Thiruvengadam. Indifferentiability of iterated Even-Mansour ciphers with non-idealized key-schedules: Five rounds are necessary and sufficient. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 524–555. Springer, Heidelberg, August 2017.

[Dut20] Avijit Dutta. Minimizing the two-round tweakable Even-Mansour cipher. In *ASIACRYPT 2020, Part I*, LNCS, pages 601–629. Springer, Heidelberg, December 2020.

[EM97] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *Journal of Cryptology*, 10(3):151–162, June 1997.

[FP15] Pooya Farshim and Gordon Procter. The related-key security of iterated Even-Mansour ciphers. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 342–363. Springer, Heidelberg, March 2015.

[GL16a] Chun Guo and Dongdai Lin. Indifferentiability of 3-round even-mansour with random oracle key derivation. *IACR Cryptol. ePrint Arch.*, page 894, 2016.

[GL16b] Chun Guo and Dongdai Lin. Separating invertible key derivations from non-invertible ones: sequential indifferentiability of 3-round even–mansour. *Designs, Codes and Cryptography*, 81(1):109–129, 2016.

[GPPR11] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES 2011*, volume 6917 of *LNCS*, pages 326–341. Springer, Heidelberg, September / October 2011.

[HT16] Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 3–32. Springer, Heidelberg, August 2016.

[ISO12] ISO/IEC. Information technology — security techniques – lightweight cryptography – part 2: Block ciphers. ISO/IEC 29192-2:2012, 2012. https://www.iso.org/standard/56552.html.

[ISO21] ISO/IEC. Information security – encryption algorithms – part 7: Tweakable block ciphers. ISO/IEC FDIS 18033-7, 2021. https://www.iso.org/standard/80505.html.

[KR07] Lars R. Knudsen and Vincent Rijmen. Known-key distinguishers for some block ciphers. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 315–324. Springer, Heidelberg, December 2007.

[LPS12] Rodolphe Lampe, Jacques Patarin, and Yannick Seurin. An asymptotically tight security analysis of the iterated Even-Mansour cipher. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 278–295. Springer, Heidelberg, December 2012.

[LS13] Rodolphe Lampe and Yannick Seurin. How to construct an ideal cipher from a small set of public permutations. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 444–463. Springer, Heidelberg, December 2013.

[ML15] Nicky Mouha and Atul Luykx. Multi-key security: The Even-Mansour construction revisited. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 209–223. Springer, Heidelberg, August 2015.

[MPS12] Avradip Mandal, Jacques Patarin, and Yannick Seurin. On the public indifferentiability and correlation intractability of the 6-round Feistel construction. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 285–302. Springer, Heidelberg, March 2012.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39. Springer, Heidelberg, February 2004.

[Pub01] NIST FIPS Pub. 197: Advanced encryption standard (aes). *Federal information processing standards publication*, 197(441):0311, 2001.

[TZ21] Stefano Tessaro and Xihu Zhang. Tight security for key-alternating ciphers with correlated sub-keys. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021, Part III*, volume 13092 of *Lecture Notes in Computer Science*, pages 435–464. Springer, 2021.

[WYCD20] Yusai Wu, Liqing Yu, Zhenfu Cao, and Xiaolei Dong. Tight security analysis of 3-round key-alternating cipher with a single permutation. In *ASIACRYPT 2020, Part I*, LNCS, pages 662–693. Springer, Heidelberg, December 2020.

[XDG23] Shanjie Xu, Qi Da, and Chun Guo. Minimizing even-mansour ciphers for sequential indifferentiability (without key schedules). In *Progress in Cryptology–INDOCRYPT 2022: 23rd International Conference on Cryptology in India, Kolkata, India, December 11–14, 2022, Proceedings*, pages 125–145. Springer, 2023.

# A Attacks on 3 Rounds

Let $\mathcal{P} = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ and $\overrightarrow{\varphi} = (\varphi_0, \varphi_1, \varphi_2, \varphi_3)$ in this section. We focus on the 3-round EM as follows.

$$\mathrm{EM}[\overrightarrow{\varphi}]_3^{\mathcal{P}}(k, u) = \varphi_3(k) \oplus \mathbf{p}_3\big(\varphi_2(k) \oplus \mathbf{p}_2\big(\varphi_1(k) \oplus \mathbf{p}_1(\varphi_0(k) \oplus u)\big)\big).$$

We distinguish two cases.

**Either $\varphi_1$ or $\varphi_2$ Has Collisions.** Wlog, assume that there exist distinct key $k, k' \in \{0,1\}^n$ such that $\varphi_1(k) = \varphi_1(k')$ while $\varphi_2(k) \neq \varphi_2(k')$ (the converse case is similar by symmetry). Then the (information theoretic) attack is as follows.

1. Pick $y_2 \in \{0,1\}^n$ in arbitrary and query $\mathbf{p}_3(y_2 \oplus \varphi_2(k)) \to y_3$ and $\mathbf{p}_3(y_2 \oplus \varphi_2(k')) \to y_3'$.
2. Query $E^{-1}(k, \varphi_3(k) \oplus y_3) \to x_1$ and $E^{-1}(k', \varphi_3(k') \oplus y_3') \to x_1'$, and output 1 if and only if $\varphi_0(k) \oplus x_1 = \varphi_0(k') \oplus x_1'$.

Clearly, it always outputs 1 when interacting with $(\mathrm{EM}_3, \mathcal{P})$. Whereas, the probability to output 1 in the ideal world is approximately $O(1/2^n)$.

**Both $\varphi_1$ and $\varphi_2$ Are (Efficient) Permutations.** Under the condition that $\varphi_1(k) \oplus \varphi_1(k') \oplus \varphi_1(k'') \oplus \varphi_1(k''') = 0$ if and only if $\varphi_2(k) \oplus \varphi_2(k') \oplus \varphi_2(k'') \oplus \varphi_2(k''') = 0$ for any four distinct keys $k, k', k'', k'''$, we observe that the attack of Andreeva et al. [ABD+13, LS13] is easily adapted to our setting (although it was described for the specific case of $\varphi_1 = \varphi_2$).

1. Pick $x_1 \in \{0,1\}^n$ in arbitrary and query $\mathbf{p}_1(x_1) \to y_1$;
2. Compute $k_1 \leftarrow \varphi_1(k)$ and $k_1' \leftarrow \varphi_1(k')$ for two distinct, arbitrarily chosen keys $k$ and $k'$;
3. Query $\mathbf{p}_2(y_1 \oplus k_1) \to y_2$, $\mathbf{p}_2(y_1 \oplus k_1') \to y_2'$, $\mathbf{p}_3(y_2 \oplus k_1) \to y_3$, and $\mathbf{p}_3(y_2' \oplus k_1') \to y_3'$;
4. Compute $k_2'' \leftarrow y_2 \oplus y_2' \oplus k_2'$ and $k_2''' \leftarrow y_2' \oplus y_2 \oplus k_2$, and further $k'' \leftarrow \varphi_2^{-1}(k_2'')$ and $k''' \leftarrow \varphi_2^{-1}(k_2''')$;
5. Query $E^{-1}(k'', \varphi_3(k'') \oplus y_3') \to u''$ and $E^{-1}(k''', \varphi_3(k''') \oplus y_3) \to u'''$, and output 1 if and only if $\varphi_0(k'') \oplus u'' = \varphi_0(k''') \oplus u'''$.

As an instance, if both $\varphi_1$ and $\varphi_2$ are *affine functions* then it does hold $\varphi_1(k) \oplus \varphi_1(k') \oplus \varphi_1(k'') \oplus \varphi_1(k''') = 0 \iff \varphi_2(k) \oplus \varphi_2(k') \oplus \varphi_2(k'') \oplus \varphi_2(k''') = 0$. This slightly strengthens existing negative results on 3 rounds.