Crooked Indifferentiability of Enveloped XOR Revisited

Rishiraj Bhattacharyya[1], Mridul Nandi[2], and Anik Raychaudhuri[2]

[1] NISER, HBNI, India, rishirajbhattacharyya@protonmail.com
[2] Indian Statistical Institute, Kolkata, India,
mridul.nandi@gmail.com,anikrc1@gmail.com

Abstract. In CRYPTO 2018, Russell,Tang, Yung and Zhou (RTYZ) introduced the notion of crooked indifferentiability to analyze the security of a hash function when the underlying primitive is subverted. They showed that the n-bit to n-bit function implemented using enveloped XOR construction (EXor) with $3n + 1$ many n-bit functions and $3n^2$-bit random initial vectors can be proven secure asymptotically in the crooked indifferentiability setting. We identify several major issues and gaps in the proof by RTYZ, We argue that their proof can achieve security only in a restricted setting. We present a new proof of crooked indifferentiability where the adversary can evaluate queries related to multiple messages. Our technique can handle function-dependent subversion.

Keywords. Crooked Indifferentiability, Subverted Random Oracle, Simulator, Enveloped XOR Hash.

1 Introduction

Blackbox Reduction and Kleptographic attack. Many of the modern cryptographic constructions are analyzed in a modular and inherently black-box manner. The schemes or protocols are built on underlying primitives only exploiting the functionality of the primitives. While analyzing the security, one shows a reduction saying, a successful attack on the construction will lead to an attack against the underlying primitive. Unfortunately, this approach completely leaves out the implementation aspects. While the underlying primitive may be well studied, a malicious implementation may embed trapdoor or other sensitive information that can be used for the attack. Moreover, such implementation may well be indistinguishable from a faithful implementation [21]. These types of attacks fall in the realm of *Kleptography*, introduced by Young and Yung [21]. While the cryptographic community did not consider kleptography as a real threat, the scenario has changed in the past few years. The kleptographic attack has been a real possibility in the post-Snowden world. A line of work has appeared aiming to formalize and provide security against kleptographic attack [2,10,17,18]. Specifically, in [2], Bellare, Paterson, and Rogaway showed that it is possible to mount algorithm substitution attacks against almost all known

symmetric key encryption schemes to the extent that the attacker learns the secret key.

Random Oracle and Indifferentiability. The *Random Oracle* methodology is a very popular platform for proving the security of cryptographic constructions in the black-box fashion. In this model, all the parties, including the adversary, are given access to a common truly random function. One proves the security of a protocol assuming the existence of such a random function. During the implementation of the protocol, the random function is replaced by a hash function H. The *Indifferentiability* framework and the composition theorem [13] assert that if the hash function H is based on an ideal primitive f, and Indifferentiable from a random function, then the instantiated protocol is as secure as the protocol in the random oracle model (assuming the security of the ideal primitive f). Indifferentiability from Random Oracle has been one of the mainstream security criteria of cryptographic hash functions. Starting from the work of Coron, Dodis, Malinaud, and Puniya [9], a plethora of results [7,4,5,12,1,6,15,14,16] have been proven, showing indifferentiability of different constructions based on different ideal primitives.

Crooked Indifferentiability. In CRYPTO 2018, Russel, Tang, Yung and Zhou [19] introduced the notion of crooked indifferentiability as a security notion for hash functions in the kleptographic setting. Like classical indifferentiability, the game of crooked-indifferentiability challenges the adversary to distinguish between two worlds. In the real world, the adversary has access to the underlying ideal primitive f, and the construction C, which has subroutine access to $\tilde{f}$, the subverted implementation of f.[3] The implementation $\tilde{f}$ on input an element x, queries the function (possibly adaptively) at maximum $\tilde{q}$ many points and, based on the transcript, decides the evaluation of x. As the adversary likes the subversion to go undetected, it is assumed that $\tilde{f}$ differs from f only on some negligible fraction (ϵ) of the domain.

In the ideal world, the construction is replaced by a Random Oracle $\mathcal{F}$. The role of f is played by a simulator with oracle access to $\mathcal{F}$ and the subverted implementation $\tilde{f}$. The job of the simulator is to simulate f in such a way that $(C^{\tilde{f}}, f)$ is indistinguishable from $(\mathcal{F}, S^{\mathcal{F},\tilde{f}})$. To avoid trivial attacks, the framework allows a *public* random string R to be used as the salt in the construction. The string R is fixed after the adversary publishes the implementation but stays the same throughout the interaction. All the parties, including the simulator and the adversary, get R as part of the initialization input. We note that even in the weaker setting of Random Oracles with auxiliary input, a random salt is required to prove security [8,11].

Enveloped Xor Construction and its crooked-indifferentiability. We recall the Enveloped XOR construction. We fix two positive integers n and l. Let $\mathcal{D} := \{0, 1, \ldots, l\} \times \{0,1\}^n$. Let H be the class of all functions $f : \mathcal{D} \to \{0,1\}^n$.

[3] The domain extension algorithms are simple and the correctness of their implementations are easy to verify.

For every $x \in \{0,1\}^n$ and an initial value $R := (r_1, \ldots, r_l) \in (\{0,1\}^n)^l$, we define

$$g_R(x) = \bigoplus_{i=1}^{l} f(i, x \oplus r_i) \quad \text{and} \quad \mathsf{EXor}(R, x) = f(0, g_R(x)).$$

In [19], Russel *et al* proved crooked-indifferentiability of the enveloped-xor construction. Their analysis is based on an interesting rejection sampling argument.

1.1 Our Contribution

Another Look at Russel et al.'s proof. We uncover that the techniques of [19], while novel and interesting, bear significant shortcomings. The consistency of the simulator is not proven. Moreover, their technical treatment requires that the subversion for the final function $f(0,\cdot)$ be independent of g_R. In other words, the proof is applicable against a restricted class of subversion. Finally, the proof does not consider the messages queried to $\mathcal{F}$. We elaborate the issues in Section 3.

A new proof of the crooked-indifferentiability of Enveloped XOR. We present a new proof of the crooked-indifferentiability of Enveloped XOR. Interestingly, our techniques do not involve heavy technical machinery. Rather, we identify core domain points related to functions and use simple tools like Markov inequality.

1.2 Overview of Our Technique.

We observe the Enveloped XOR (EXoR) construction is in the class of Generalized Domain Extensions considered in [5]. It is known that for a GDE construction with independent functions and preimage awareness, the indifferentiability advantage is bounded by the probability that the final chaining query is not fresh. However, EXoR construction instantiated with the crooked functions (denoted by $\widetilde{\mathsf{EXor}}$) is not part of GDE. The main issue is that the final output of $\widetilde{\mathsf{EXor}}$ need not be the output of $f(0,\cdot)$ evaluation, as required by the condition of GDE.

We consider an intermediate construction $\overline{\mathsf{EXor}}(R, m) = f(0, \tilde{g}_R(m))$. In other words, the intermediate construction restricts that the finalization function $f(0,\cdot)$ is not subverted. $\overline{\mathsf{EXor}}$ is a GDE construction and crooked-indifferentiability of $\overline{\mathsf{EXor}}$ can be proved following the structure of [5]. In particular, the generic simulator of [5], adopted for $\overline{\mathsf{EXor}}$ along with access to $\tilde{f}$ work out here along with the consistency arguments. Our proof is modularized via the following two claims.

– Claim 3 shows distinguishing advantage for $(f, \overline{\mathsf{EXoR}})$ and $(f, \widetilde{\mathsf{EXoR}})$ is bounded by probability of hitting a crooked point or domain point for f_0 (Bad1).

– Claim 4 shows the distinguishing advantage of intermediate world $(f, \overline{\mathsf{EXoR}})$ and the ideal world of crooked indifferentiability is bounded by the probability of Bad2 event. This event is classified into two main categories. In the first category, while responding to a query to the primitive (or the simulator), input $\tilde{g}_R(m)$ appeared already in the transcript. In the second category, the input $\tilde{g}_R(m)$ appeared in the extended transcript which includes all queries of a subverted computation $\tilde{f}(x)$ of a crooked point x.

The challenge remaining is to bound the probability of the bad events. Our proof works with a counting approach. We say a point $\alpha \in \{0,1\}^n$ is robust with respect to a function f, if all points which queries α is not subverted with all but negligible probability, if the output $f(\alpha)$ is re-sampled. A point is good if it is queried by only a few robust and un-crooked points. By averaging argument, we show that for overwhelming fraction of candidate f, R, for every message m, there will exist an index i such that $m \oplus r_i$ is good for function $f(i, \cdot)$. Now, we can say that even though $f(i, m \oplus r_i)$ was queried by other points, they are robust. If we re-sample at $(i, m \oplus r_i)$ the subverted outputs of those robust points will not change. Thus, we can talk about $\tilde{g}_R(m)$ independently to the outputs of the function $\tilde{f}(0, \cdot)$.

Finally, we shall show that the output distribution of $\tilde{g}_R(m)$ is close to uniform. We could find a rejection resampling lemma on two or more points, and argue the uniformity of $\tilde{g}_R(m)$. However, we simplify things further. We observe that with high probability over the output values of $f(i, m \oplus r_i)$ for every i for which $m \oplus r_i$ is good in f, the transcript of the previous internal queries remains unchanged. Hence, we consider the conditional probabilities by conditioning on all possible transcripts and take union bound to show near uniformity of $\tilde{g}_R(m)$.

Relation of GDE constructions with Our Results and Further Uses. A majority of this work focuses on $\overline{\mathsf{EXor}}$ construction which is a GDE construction (defined in [5]). GDE constructions cover a wide range of domain extension algorithms. We believe that many ideas developed in this result to deal with the $\overline{\mathsf{EXor}}$ construction can be extended to investigate the crooked-indifferentiability of different GDE constructions. However the bad events and their analysis will depend on the particular construction being investigated.

Revised Proof by Russel et al. After we communicated our findings to the authors of [19], they acknowledged the issues, and uploaded a major revision in eprint [20]. Our proof is done independently and significantly differ from their revised proof in some crucial aspects.

2 Notations and Preliminaries

NOTATIONS. Let $\mathbb{N} = \{0, 1, \ldots\}$ be the set of natural numbers. If $k \in \mathbb{N}$, then $\{0,1\}^k$ denotes the set of all k-bit binary strings. If x and y are two strings

xy denote the concatenated string.We write $x \xleftarrow{\$} S$ to denote the process of choosing x uniformly at random from a set S and independently from all other random variables defined so far. For a positive integer l, we use $(l]$ and $[l]$ to denote the set $\{1, \cdots, l\}$ and $\{0, 1, \ldots, l\}$ respectively. The positive integer n is our security parameter and we write $\mathcal{R} := \{0,1\}^n$.

Class of Functions. $\mathsf{H} := \mathsf{H}_{\mathcal{D},n}$ denotes the set of all n-bit functions from $\mathcal{D}$ to $\mathcal{R}$. In this paper we mainly consider $\mathcal{D} :=: [l] \times \{0,1\}^n$ and let $f : [l] \times \{0,1\}^n \to \mathcal{R}$ denotes a family of l many functions from $\{0,1\}^n$ to itself. *We often use the shorthand f to denote the family $\{f_0 := f(0,\cdot), \cdots, f_l := f(l,\cdot)\}$ when the function family is given as oracles.*

For any tuples of pairs $\tau = ((x_1, y_1), \ldots, (x_s, y_s))$ we write $\mathcal{D}(\tau)$ (called domain of τ) to denote the set $\{x_i : 1 \le i \le s\}$. We say a function f agrees with τ if for all $(x, y) \in \tau$, $f(x) = y$. For every $x \in \mathcal{D}$, $\alpha \in \mathcal{R}$, we use $f|_{x\to\alpha}$(or simply f_α whenever x is understood) to denote the following function:

$$f|_{x\to\alpha}(y) = \begin{cases} f(y) & \text{if } x \neq y \\ \alpha & \text{if } x = y \end{cases}.$$

Adversaries and Distinguishing Advantage. An *adversary* $\mathtt{A}$ is an algorithm possibly with access to oracles $\mathcal{O}_1, \ldots, \mathcal{O}_\ell$ denoted by $\mathtt{A}^{\mathcal{O}_1,\ldots,\mathcal{O}_\ell}$. The adversaries considered in this paper are computationally unbounded. The complexities of these algorithms are measured solely on the number of queries they make. An algorithm $\mathcal{A}$ having access to an oracle is called q-query algorithm if it makes at most q queries to its oracle. Similarly, an oracle algorithm having access to two oracles is called (q_1, q_2)-query algorithm, if it makes at most q_1 and q_2 queries to its first and second oracles respectively. Adversarial queries and the corresponding responses are stored in a transcript τ. So, $\mathcal{D}(\tau)$ denotes the list of inputs (queries) in the transcript.

Definition 1 (Distinguishing Advantage). *Let F^l and G^l be two l-tuples of probabilistic oracle algorithms for some positive integer l. We define* advantage *of an adversary $\mathcal{A}$ at distinguishing F^l from G^l as*

$$\Delta_{\mathcal{A}}(F^l \ ; \ G^l) = \left|\Pr[\mathcal{A}^{F_1,F_2,\cdots,F_l} = 1] - \Pr[\mathcal{A}^{G_1,G_2,\cdots,G_l} = 1]\right|.$$

If $\mathcal{A}$ makes a total of q queries, it is called a q-query distinguisher.

2.1 Modeling Subversion Algorithms and Crooked-Indifferentiability

We recall the related terms and notations introduced in [19] in our terminologies. A $(q, \tilde{q})$ *implementor* is an q-query oracle algorithm $\mathcal{A}^{\mathcal{O}}$. $\mathcal{A}$ outputs the description of another oracle algorithm $\tilde{F}^{\mathcal{O}}$ (called subversion algorithm) which makes at most $\tilde{q}$ many queries to its oracle. For a correct subversion algorithm of a function $f \in \mathsf{H} := \mathsf{H}_{\mathcal{D},n}$, we must have $\tilde{f} := \tilde{F}^f = f$. However, a crooked-implementor may not behave correctly.

Definition 2 (crooked implementer). *A $(q, \tilde{q})$ implementer $\mathcal{A}_1$ is called* ϵ-crooked *for a function family* H*, if for every* $f \in \mathsf{H}$ *and* $\tilde{f} \leftarrow \mathcal{A}_1^f$*,*

$$\Pr_{\alpha \xleftarrow{\$} \mathcal{D}} [\tilde{f}(\alpha) \neq f(\alpha)] \leq \epsilon.$$

Let τ_0 denote the transcript of oracle queries of $\mathcal{A}_1^f$. We may assume that ϵ is negligible[4] and the transcript τ_0 is hardwired in $\tilde{f}$ and all the $\tilde{q}$ queries made by $\tilde{f}$ are different from $\mathcal{D}(\tau_0)$ (as the response is known from the transcript). The subversion algorithm $\tilde{f}$ on input x queries $\gamma_1(x), \gamma_2(x), \ldots, \gamma_{\tilde{q}}(x)$, and based on the query-responses outputs $\tilde{f}(x)$. Without loss of generality, we assume $\gamma_1(x) = x$. We write $Q(x) := Q_f(x)$ to denote the set of all queries as defined above. We write

$$Q_f^{-1}(y) := \{x : y \in Q_f(x)\},$$

the set of all points x, in which the computation of $\tilde{f}(x)$ queries y. The set does not depend on the value of $f(x)$. Mathematically, let $f_\beta = f|_{y \to \beta}$ then $Q_f^{-1}(y) = Q_{f_\beta}^{-1}(y)$ for all β.

Crooked Indifferentiability. A crooked distinguisher is a two-stage adversary; the first stage is a subverted implementor and the second stage is a distinguisher.

Definition 3 (crooked distinguisher). *We say that a pair* $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ *of probabilistic algorithms* $((q_1, \tilde{q}, \epsilon), q_2)$-crooked distinguisher *for* H *if*

(i) $\mathcal{A}_1$ *is a* ϵ*-crooked* $(q_1, \tilde{q})$ *implementer for* H *and*

(ii) $\mathcal{A}_2(r, \tau_0, \cdot)$ *is a* q_2*-query distinguisher where* r *is the random coin of* $\mathcal{A}_1$*, and* τ_0 *is the advice-string, the transcript of interaction of* $\mathcal{A}_1$ *with* f*.*

Note that the random coin r of $\mathcal{A}_1$ and the transcript of $\mathcal{A}_1$ are shared with $\mathcal{A}_2$ to emphasis that $\mathcal{A}_1$ and $\mathcal{A}_2$ are joint adversary working in two different stages.

Definition 4 (H-crooked-indifferentiability [19]). *Let* $\mathcal{F}$ *be an ideal primitive and* C *be an initial value based* $\mathcal{F}$*-compatible oracle construction. The construction* C *is said to be* $((q_1, \tilde{q}), (q_2, q_{\text{sim}}), \epsilon, \delta)$**-crooked-indifferentiable from** $\mathcal{F}$ *if there is a* q_{sim}*-query algorithm* S *(called simulator) such that for all* $((q_1, \tilde{q}, \epsilon), q_2)$*-crooked distinguisher* $(\mathcal{A}_1(r), \mathcal{A}_2(r, \cdot, \cdot))$ *for* H*, we have*

$$\Delta_{\mathcal{A}_2(r,\tau_0,R)}\big((f, C^{\tilde{f}}(R, \cdot)) \; ; \; (S^{\mathcal{F},\tilde{F}}(\tau_0, R), \mathcal{F})\big) \leq \delta \tag{1}$$

where τ_0 *is the advice string of* $\mathcal{A}_1^f$ *and* R *is the random initial value of the construction sampled after subverted implementation is set.*

[4] Given an implementation, one may check the correctness of the algorithm by comparing the outputs of the implementation with a known correct algorithm. More precisely, we sample $\alpha_1, \ldots, \alpha_t \xleftarrow{\$} \{0,1\}^m$ and then for all $0 \leq i \leq l$, we check whether $\tilde{f}(\alpha_i) = f(\alpha_i)$ holds. If it does not hold, the implementation would not be used. It is easy to see that for ϵ-crooked implementation the subversion would not be detected with probability at least $(1 - \epsilon)^t$. So for a negligible ϵ, this probability would be still close to one for all polynomial function t and so the implementation can survive for further use.

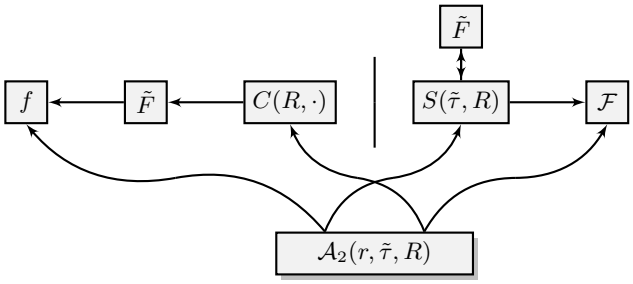

Fig. 1: The crooked-indifferentiability notion. In the first phase of real world, $\mathcal{A}_1$ interacts with f and returns an oracle algorithm $\tilde{F}$ (which would be accessed by the construction C in the second phase). In the second phase the random initial value R will be sampled and given to construction C and also to $\mathcal{A}_2$. In ideal world, simulator $S^{\mathcal{F}}$ gets the transcript of the first phase as advice string, blackbox access to the subverted implementation $\tilde{F}$ and the initial value R.

Two-Stage Distinguishing Game. Now we explain the distinguishing game. In the first stage, $\mathcal{A}_1^f$ outputs $\tilde{F}$ after interacting with a random oracle f as discussed before. A random initial value R, for the hash construction C is sampled. In the real world, $\mathcal{A}_2$ interacts with the same f of the first stage and the construction $C^{\tilde{f}}(R,\cdot)$. In the ideal world, the simulator S gets the advice-string τ_0, the initial value R and blackbox access to the subverted implementation $\tilde{F}$ and a random oracle $\mathcal{F}$. Simulator is aimed to simulate f so that behavior of $(f, C^{\tilde{f}})$ is as close as $(S,\mathcal{F})$ to the distinguisher $\mathcal{A}_2$.

Convention on Crooked Distinguishers: Note that there is no loss to assume that both $\mathcal{A}_1$ and $\mathcal{A}_2$ are deterministic (so we skip the notation r) when we consider computational unbounded adversary[5]. We also assume that $\mathcal{A}_2$ makes all distinct queries and distinct from the queries made by $\mathcal{A}_1$ (as the simulator has the advice string and so it can respond honestly). We skip the notation τ_0 as an input of $\mathcal{A}_2$ as it is fixed throughout the game. As the advice string is fixed, we consider it as part of the transcript. Specifically, the transcript τ_0, view of $\mathcal{A}_2$ at the start of the second stage is set as the advice string τ_0. We fix the advise string τ_0 throughout the paper. We write f to denote the random function agreeing on τ_0. In other words, $\mathsf{f} \xleftarrow{\$} \Gamma_{\tau_0} = \{f : f(x) = y, (x,y) \in \tau_0\}$.

Enveloped XOR Construction. Recall that, in the real world, the distinguisher is interacting with the subverted construction $\widetilde{\mathsf{EXor}}$ which is defined as

$$\widetilde{\mathsf{EXor}}(R,M) = \tilde{f}(0,\tilde{g}_R(M)) \quad \text{where} \quad \tilde{g}_R(M) = \bigoplus_{i=1}^{l} \tilde{f}(i, M \oplus r_i).$$

[5] $\mathcal{A}_1$ can fix the best random coin for which the distinguishing advantage of $\mathcal{A}_2$ is maximum.

We also define a hybrid construction $\overline{\mathsf{EXor}}[f](R, M) = f(0, \tilde{g}_R(M))$. Now consider an adversary $\mathcal{A}$ interacting with $(f, \overline{\mathsf{EXor}} := \overline{\mathsf{EXor}}[f])$.

Assumption on Adversary. For all primitive queries of the form (j, x) with $j > 0$, we return $\overline{\mathsf{EXor}}(m)$ and all responses of all queries $(a, \alpha_a), a \in [l]$ where $\alpha_a = m + R_a$ and $m = x + R_j$. Note that simulator can compute m and so responding $\overline{\mathsf{EXor}}(m)$ honestly for simulator would not be a problem. Moreover, we assume that adversary disclose all queries for the construction to the simulator.

Transcript of Interaction. For $j \geq 0$, let $\mathsf{Tr}_j := (R, \tau_j, \pi_j)$ denote the transcript (random variable due to randomness of f only) of $\mathcal{A}$ after j queries where R is the initial value of the construction, and τ_j, π_j denote the query-responses for the primitive and the construction respectively. Note that τ_j contains τ_0 for all j.

3 Revisiting the Crooked Indifferenitability Security of EXoR [19].

A brief detour: classical indifferentiability simulator for EXor. Before describing the crooked indifferentiability simulator, we would like to briefly recall the principle behind the indifferentiability simulator and proof principles behind EXor construction in the classical setting.

The goal of the simulator is to simulate each $f(i, \cdot)$ honestly so that for every queried message m, it holds that $\mathsf{EXor}(R, m) = \mathcal{F}(m)$ for all queried m. Without loss of generality, assume that whenever the adversary makes queries $f(i, x)$ for $i > 0$, it also makes queries $f(j, x \oplus r_i \oplus r_j)$ for all $j > 0$ simultaneously. In other words, it makes a batch query of the form $(f(j, m \oplus r_j))_{1 \leq j \leq l}$ for some $m \in \{0,1\}^n$. We simply say that the adversary $\mathcal{A}$ queries m to g_R and obtains responses $(f(j, m \oplus r_j))_{1 \leq j \leq l}$.

On receiving a batch query $g_R(m)$, the simulator will honestly sample outputs for the corresponding $f(i, m \oplus R_i)$ queries for all $i \in (l]$, and compute $g_R(m)$ by xoring those sampled outputs. Also, the simulator will save the queried m along with the computed $g_R(m)$ in a list L. For a $f(0, x)$ query, the simulator will first search in L, whether for some m, it has given $x = g_R(m)$ as output. If yes, the simulator simply returns $\mathcal{F}(m)$. If no such entry exists, the simulator samples an output z uniformly at random and returns z.

Now, we briefly recall how the indifferentiability is proved for this simulator. There are two bad events.

- for distinct m, m', it holds that $g_R(m) = g_R(m')$. In this case, the simulator, on query $f(0, g_R(m))$ can not be consistent with both $\mathcal{F}(m)$ and $\mathcal{F}(m')$ with any significant probability.
- For a batch query $g_R(m)$ the output is such that it matches with a previous $f(0, .)$ query. In this case, the simulator has already given output to the $f(0, .)$ query which, with all but negligible probability, is not equal to $\mathcal{F}(m)$.

One can indeed summarize these bad events as one; $g_R(m) \in E$, where E is the set of $f(0, .)$ queries made by the adversary.

The Simulator for Crooked Indifferentiability. We now describe the main indea behind the simulator in the crooked indifferentiability setting. The same principle was used in [19]. Note, here the main goal of the simulator is different. It needs to simulate $f \xleftarrow{\$} \mathsf{H}$ as honestly [6] as possible such that $\widetilde{\mathsf{EXor}}(R, m) = \mathcal{F}(m)$ for all queried m. Thus the simulator needs to ensure that the output of the random oracle matches with the *subverted implementation* of EXor.

The simulator maintains a list L of pairs (α, β) to record $f(\alpha) = \beta$ for $\alpha \in \mathcal{D}$ and $\beta \in \{0,1\}^n$. It also maintains a sub-list $L^A \subseteq L$ consisting of all those pairs which are known to the distinguisher. Both lists are initialized to z (the advice-string in the first stage which we fix to any tuple of q_1 pairs). $L_0 = L_0^A = z$. Now we describe how the simulator responds.

1. (Query $f(0, w)$) We call this query a Type-1 Query. Type-1 Queries are returned honestly. If $((0, w), y) \in L$ for some y, the simulator returns the same y. Otherwise, it samples y uniformly from $\{0,1\}^n$, updates the list L and L^A, and returns y.
2. (Query $g_R(m)$) We call this Type-2 Query. For a query $g_R(m)$ (i.e. batch query) the simulator computes $\tilde{f}(\alpha_j)$ for all j, one by one by executing the subverted implementation $\tilde{F}$, where $\alpha_j = (j, m \oplus R_j)$. During this execution, simulator responds honestly to all queries made by the subverted implementation and updates the L-list by incorporating all query responses of h. However, it updates L^A list only with $(\alpha_j, f(\alpha_j))$ for all j. Let $\tilde{\mathsf{g}} := \bigoplus_j \tilde{f}(\alpha_j)$. If $(0, \tilde{\mathsf{g}}) \in \mathcal{D}(L)$, the simulator **aborts** . If the simulator does not abort, it makes a query $\mathcal{F}(m)$ and adds $((0, \tilde{\mathsf{g}}), \mathcal{F}(m))$ into the both lists L and L^A.

For $f(0, w)$ made by $\mathcal{A}_2$ where $w = \tilde{g}_R(m)$ for some previous query m to g_R, the simulator responds as $\mathcal{F}(m)$.

Cautionary Note. Even though $\mathcal{F}$ is a random oracle, we cannot say that the probability distribution of the response of $(0, \tilde{\mathsf{g}})$ in the ideal world is uniform. Note that, the adversary can choose m after making several consultations with $\mathcal{F}$. In other words, m can be dependent on $\mathcal{F}$. For example, the adversary can choose a message m for which the last bit of $\mathcal{F}(m)$ is zero. Thus, the response for the query $(0, w)$ always has zero as the last bit (which diverts from the uniform distribution). However, the randomness can be considered when we consider joint probability distribution of all query-responses.

Transcript: Now we describe what is the transcript to the distinguisher and for the simulator in more detail. First, we introduce some more relevant notations.

1. Let L^F denote the set of all pairs $(m', \tilde{z})$ of query response of $\mathcal{F}$ by $\mathcal{A}_2$.
2. Let L^g denote the set of all pairs (m, β^l) of query response of g_R oracle (batch query) made by $\mathcal{A}_2$ to the simulator where $\beta^l := (\beta_1, \ldots, \beta_l)$ and

[6] By honestly we mean perfectly simulating a random function. If the responses are already in the list it returns that value, otherwise, it samples a fresh random response and includes the input and output pairs in the list.

$\beta_j = h(j, m \oplus R_j)$ for all j. According to our convention all these m must be queried to $\mathcal{F}$ beforehand.

3. As we described, we also have two lists, namely L and its sublist L^A, keeping the query responses of h oracle.

Now we define the transcript and partial transcript of the interaction. We recall that q_1 is the number of queries in the first stage and $\mathcal{A}_2$ is a (q_F, q_2)-query algorithm. Let $q = q_2 + q_F$ For any $1 \leq i \leq q$, we define the partial transcript of $\mathcal{A}$ and the simulator as $\tau_i^A := (L_i^F, L_i^A)$ and $\tau_i^S := (L_i, L_i^g)$ respectively, where L_i^F, L_i^A, L_i, L_i^g denote the contents of the corresponding lists just before making ith query of the distinguisher. So when, $i = 1$, $L_1^A = L_1 = z$ and the rest are empty and when $i = q + 1$, these are the final lists of transcripts. Let $\tau_i := (\tau_i^A, \tau_i^S)$ and $\tau := (\tau^A, \tau^S)$ denote the joint transcript on ith query or after completion respectively. As the adversary is deterministic, the simulator is also deterministic for a given h and $\mathcal{F}$, and we have fixed z, a (partial) transcript is completely determined by the choice of R, h and $\mathcal{F}$ (in the ideal world). We write $(R, f, \mathcal{F}) \vdash \tau_i^S$ if the transcript τ_i^S is obtained when the initial value is R, the random oracles are $\mathcal{F}$ and f. We similarly define $(R, f, \mathcal{F}) \vdash \tau_i^A$ and $(R, f, \mathcal{F}) \vdash \tau_i$.

3.1 Techniques of [19]

Overview of the techniques in [19]. We assume, without any loss of generality that the second stage adversary $\mathcal{A}_2$ queries m to $\mathcal{F}$ before it queries to g_R oracle. In addition, like before, we assume that it makes batch queries.

For every query number i, we define a set $E_i := \mathcal{D}(L_i) \cup \mathsf{subv}_f$ where subv_f is the set of all crooked elements for f. The event $\textsc{Bad}_i$ holds if and only if $(0, \tilde{g}_R(m_i)) \in E_i$ where m_i denotes the ith query of $\mathcal{A}$ (made to g_R oracle of the simulator). So, the crooked indifferentiable advantage is bounded by $\sum_{i=1}^{q_2} \Pr(\tilde{g}_R(m_i) \in E_i)$. The authors wanted to show that the distribution of $\tilde{g}_R(m_i)$ is almost uniform. They proposed the following theorem.

(**Theorem 5** from [19]). With overwhelming probability (i.e., one minus a negligible amount) there exists a set $\mathcal{R}_{\tau_0} \subseteq (\{0,1\}^n)^l$ and for every i, a set of transcripts $\mathcal{T}_i^A$ (before ith query) such that for all $R \in \mathcal{R}_{\tau_0}$, $\tau_i := (L_i^F, L_i^A) \in \mathcal{T}_i^A$, and $m \notin \mathcal{D}(L_i^g)$,

$$\Pr_{\mathsf{f}}[(0, \tilde{g}_{\mathsf{R}}(m)) \in E_i \mid (R, \mathsf{f}, \mathcal{F}) \vdash \tau_i] \leq \mathrm{poly}(n)\sqrt{|E_i|} + \mathsf{negl}(n).$$

The authors claimed that crooked indifferentiability of EXor can be derived from the above theorem. To describe the issues we need to dive into the main idea which is to show that $\tilde{g}_{\mathsf{R}}(m)$ behaves close to the uniform distribution over $\{0,1\}^n$. Thus the above probability would be negligible as $q_1/2^n$ and $|\mathsf{subv}_f|/2^n$ is negligible. By using Markov inequality, authors are able to identify a set of overwhelming amount of pairs (R, f), called *unpredictable* pair, such that for any unpredictable (R, f) all m, there exists an index i such that

1. $\Pr_\beta[\alpha_i \in \mathsf{subv}_f \mid f(\alpha_i) = \beta]$ is negligible and
2. $\alpha_j \notin Q_f^{-1}(\alpha_i)$ for all $j \neq i$, where $\alpha_j = m \oplus R_i$.

Thus, if we resample $\beta = f(\alpha_i)$ then with overwhelming probability $\tilde{f}|_{\alpha_i \to \beta}(\alpha_i) = f|_{\alpha_i \to \beta}(\alpha_)$ (i.e. α_i is not crooked and returned a random value) and all corresponding values for indices j different from i will remain same. So, $\tilde{g}_R(m) = \beta + A$ where A does not depend on choice of β. Thus, the modified distribution is close to uniform (as almost all values of β will be good). In particular the authors made the following claim:

Claim 1 *Under the modified distribution (i.e. after resampling),* $\Pr[\tilde{g}_R(m) \in E_1] \leq q_1/2^n + \epsilon + p_n$ *where* p_n *denotes the probability that a random pair* (R, f) *is not unpredictable.*

As the choice of i depends on the function f and so a new rejection resampling lemma is used to bound the probability of the event under the original distribution (i.e. before resampling).

Lemma 1 (Rejection Resampling [19]). *Let* $X := (X_1, \ldots, X_k)$ *be a random variable uniform on* $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_k$. *Let* $A : \Omega \to (k]$ *and define* $Z = (Z_1, \ldots, Z_k)$ *where* $Z_i = A_i$ *except at* $j = A(X^k)$ *for which* Z_j *is sampled uniformly and independently of remaining random variables. Then for any event* $S \subseteq \Omega$, *it holds that*

$$|S|/|\Omega| \leq \sqrt{k \Pr(Z \in S)}$$

With this rejection resampling result and the Claim 1, the authors concluded the following under original distribution:

$$\Pr_{h*}(\tilde{g}_R(x) \in E_1) \leq \sqrt{l \cdot \Pr_{\text{resampled } h}(\tilde{g}_R(x) \in E_1)} \leq \sqrt{l \cdot (q_1/2^n + \epsilon + p_n)}.$$

3.2 Issues with the technique of [19]

Now we are ready to describe the issues and the limitations of the techniques in [19]. To prove the general case (i.e. for any query), the authors provide a proof sketch where they argued that with an overwhelming probability of realizable transcript $\mathcal{T}$ and for all $\tau \in \mathcal{T}$, $\Pr(\tilde{g}_R(m_i) \in E_i \mid \tau)$ is negligible.

The number of queries to $\mathcal{F}$ is essential. An incompleteness of the proof of [19] comes from the fact that the analysis does not consider the $\mathcal{F}$ queries of the distinguisher. The bound is almost vanishing if $q_1 = 0$ and $q_2 = 2$ and there is no crooked point. However, a distinguisher can search for $m \neq m'$ such that $\mathcal{F}(m) = \mathcal{F}(m')$. Conditioned on collision at the final output, the event $g_R(m) = g_R(m')$ holds with probability about $1/2$. On the other hand, for the honest simulation of all f values, g value will collide with very low probability. If the adversary can make $2^{n/2}$ many queries to $\mathcal{F}$, the above inconsistency can be forced. Hence, *the probability upper bound of Theorem 5 of [19] can not be independent of the number of queries made to* $\mathcal{F}$.

Inconsistency for Multiple Queries: Controlling query dependencies for the same index. Authors claimed that for all unpredictable (R, h), for all m, an index i exists on which the resampling can be done *without affecting the transcript*. Recalling the notion of unpredictable (R, h) we see that the resampling is done on an index i, that is honest ($\tilde{f}(i, m \oplus R_i) = f(i, m \oplus R_i)$, and $f(i, m \oplus R_i)$ is not queried by $f(j, m \oplus R_j)$ for any other j. From here, the authors argued that the transcript of the interaction remains same, if we resample at such i. This claim is justified for a single message and not for multiple queries. We note that it is easy to construct a subverted implementation $\tilde{F}$ for which all inputs of f for a batch response are queried during some other previous query. For example, if it queries $f(i, x \oplus 1^n)$ for an input (i, x), and the distinguisher makes two batch queries queries, $\tilde{g}_R(m \oplus 1^n)$, and $\tilde{g}_R(m)$. The simulator, while simulating $\tilde{g}_R(m \oplus 1^n)$ responds to all the queries made by $\tilde{f}(i, m \oplus 1^n \oplus R_i)$, and in particular the value of $f(i, m \oplus R_i)$ is now gets fixed. *So an appropriate analysis was missing in case of multiple queries.*

The bad event E_i depends on the function f. The main technical claim of [19] that $\Pr_{\text{resampled } f}(\tilde{g}_R(x) \in E)$ is small because $\tilde{g}_R(x)$ is uniformly distributed under resampling distribution of f and size of E is negligibly small. However the crooked set of $f(0, \cdot)$ may depend on the other functions $f(1, \cdot), \ldots, f(\cdot)$. Thus the event E is not independent of $\tilde{g}_R(x)$. In particular, one cannot upper bound the $\Pr(\tilde{g}_R(x) \in E)$ as $|E|/2^n$. This is one of the crucial observation which actually makes the crooked security analysis quite a complex task.

4 Basic Setup: Good Pairs and Critical Set

Subverted inputs. For a function $f\colon \mathcal{D} \to \mathcal{R}$ agreeing on τ_0, we define

$$\mathsf{subv}_f = \{x \mid x \in \mathsf{Dom}(\tau_0) \vee \tilde{f}(x) \neq f(x)\},$$

union of the set of all subverted points for the function f and the $\mathsf{Dom}(\tau_0)$. We consider elements of the domain of τ_0 as subverted points as the outputs of those have no entropy and is hard coded into an implementation. Thus, we treat all those inputs as subverted points. Clearly, for all function f,

$$|\mathsf{subv}_f| \leq q_1 + \epsilon|\mathcal{D}|.$$

where q_1 denotes the size of τ_0. Let $\epsilon_1 := \epsilon + q_1/|\mathcal{D}|$.

Definition 5 (robust point). *Let f agree on τ_0. A point y is called* robust *in f (or the pair (y, f) is called robust) if for all $x \in Q_f^{-1}(y)$,*

$$\Pr_{\beta}\left[x \in \mathit{subv}_{f_\beta}\right] \leq \sqrt{\epsilon_1}$$

where $\beta \xleftarrow{\$} \mathcal{R}$ and $f_\beta := f|_{y \to \beta}$.

Note that robustness of y in f does not depend on the value $f(y)$. In other words, if y is robust in f then so in $f|_{y \to \beta}$ for all β.

Definition 6 (popular point). *A point $y \notin \mathsf{Dom}(\tau_0)$ is called* popular *for a function f if $|Q_f^{-1}(y)| > \epsilon_1^{-1/4}$.*

Recall that the subversion algorithm $\tilde{f}$ makes at most $\tilde{q}$ many queries for any y. So, $\sum_y |Q_f^{-1}(y)| \leq \tilde{q}|\mathcal{D}|$. Using the simple averaging argument the number of popular points are at most $\tilde{q}\epsilon_1^{\frac{1}{4}}|\mathcal{D}|$.

$$\Pr_{x,f}[x \text{ is popular in } f] \leq \tilde{q}\epsilon_1^{\frac{1}{4}} \tag{2}$$

We call the robust pair (y, f) **good** if (1) y is not popular for f and (2) for all $x \in Q_f^{-1}(y)$, $x \notin \mathsf{subv}_f$. In particular for good (y, f), it holds that $y \notin \mathsf{subv}_f$ and $y \notin \mathsf{subv}_{f_\beta}$ with high probability over randomness of β where $f_\beta := f|_{y \to \beta}$.

Lemma 2. *For a random $y \xleftarrow{\$} \mathcal{D}$, we have*

$$\Pr_{y,\mathsf{f}}[(y, \mathsf{f}) \textit{ is not good}] \leq 3\tilde{q}\epsilon_1^{\frac{1}{4}}.$$

Proof. We define two indicator functions:

$$d(x, f) = \begin{cases} 1, & \text{if } x \in \mathsf{subv}_f \\ 0, & \text{otherwise} \end{cases} \qquad d_{j,\beta}(x, f) = \begin{cases} 1, & \text{if } x \in \mathsf{subv}_{f|_{\gamma_j^{(x)} \to \beta}} \\ 0, & \text{otherwise.} \end{cases}$$

In other words, $d(x, f)$ simply indicator function for capturing crooked points and $d_{j,\beta}(x, f)$ is an indicator function capturing whether a point x becomes crooked for f after replacing the jth query output by β. For $1 \leq j \leq \tilde{q}$, let $D^j(x, f) = \mathbb{E}_\beta(d_{j,\beta}(x, f))$. For any function $g \in \Gamma_{\tau_0}$, let $\mathcal{S}_{x,g} := \{(f, \beta) : f|_{\gamma_j^{(x)} \to \beta} = g\}$. It is easy to see that we have $|\mathcal{S}_{x,g}| = 2^n$. Now, for each j,

$$\begin{aligned}
\mathop{\mathbb{E}}_{x,\mathsf{f}}\left(D^j(x, \mathsf{f})\right) &= \mathop{\mathbb{E}}_{x,\mathsf{f}} \mathop{\mathbb{E}}_{\beta}\left(d_{j,\beta}(x, \mathsf{f})\right) \\
&= \sum_{x,\mathsf{f},\beta} \Pr(\mathsf{f}) \Pr(x) \Pr(\beta) \cdot d_{j,\beta}(x, \mathsf{f}) \\
&= 2^{-n} \sum_{(f,\beta) \in \mathcal{S}_{x,g}} \sum_{x,g} \Pr(g) \Pr(x) \cdot d(x, g) \\
&= \sum_{x,g} \Pr(g) \Pr(x) \cdot d(x, g) \\
&= \mathop{\mathbb{E}}_{x,g} d(x, g) \leq \epsilon + \frac{q_1}{|\mathcal{D}|} := \epsilon_1
\end{aligned}$$

Applying Markov inequality, we get for every $j \in (\tilde{q}]$

$$\Pr_{x,\mathsf{f}}\left[D^j(x, \mathsf{f}) \geq \epsilon_1^{\frac{1}{2}}\right] \leq \frac{\mathbb{E}_{x,\mathsf{f}}\left(D^j(x, \mathsf{f})\right)}{\epsilon^{\frac{1}{2}}} \leq \epsilon_1^{\frac{1}{2}} \tag{3}$$

We recall there are three ways x can be not good in f.

$$\Pr_{\mathsf{f},x}[(x,\mathsf{f}) \text{ is not good}] \leq \Pr_{\mathsf{f},x}[x \text{ is popular for } \mathsf{f}] + \Pr_{\mathsf{f},x}[x \text{ is queried by some point in } \mathsf{subv}_{\mathsf{f}}] + \Pr_{\mathsf{f},x}[(x,\mathsf{f}) \text{ is not robust} \mid x \text{ is not popular for } \mathsf{f}]$$

As there are at most $\epsilon_1|\mathcal{D}|$ many points in subv_f,

$$\Pr_{\mathsf{f},x}[x \text{ is queried by some point in } \mathsf{subv}_{\mathsf{f}}] \leq \tilde{q}\epsilon_1.$$

From the definition of robust points and Equation 3

$$\begin{aligned}\Pr_{x,\mathsf{f}}[x \text{ is non robust in } \mathsf{f} \mid x \text{ is not popular for } \mathsf{f}] &\leq \epsilon_1^{-1/4} \sum_{j=1}^{\tilde{q}} \Pr_{x,\mathsf{f}}\left[D^j(x,\mathsf{f}) \geq \epsilon_1^{\frac{1}{2}}\right] \\ &\leq \tilde{q}\epsilon_1^{\frac{1}{4}}\end{aligned}$$

Adding above two inequalities and Equation 2

$$\Pr_{\mathsf{f},x}[x \text{ is not good in } \mathsf{f}] \leq \tilde{q}\left(\epsilon_1 + \epsilon_1^{\frac{1}{4}} + \epsilon_1^{\frac{1}{4}}\right) \leq 3\tilde{q}\epsilon_1^{\frac{1}{4}}$$

□

Critical Set. We consider a set $\mathcal{G}$ of pairs (R, f) of initial values R and functions f satisfying the condition that for every $m \in \{0,1\}^n$ there exists $1 \leq i \leq l$ such that $(\alpha_i := (i, m \oplus R_i), f)$ is good. The following lemma says that for a uniform random string R (initial value) and a randomly chosen function f agreeing on τ_0, with high probability (R, f) is in the critical set.

Lemma 3. *Let $\tilde{q} \leq 2^{n/2}, \epsilon_1 \leq \frac{1}{2^{16}}$ and $\ell > 2n$. It holds that*

$$\Pr_{R,\mathsf{f}}((R,\mathsf{f}) \notin \mathcal{G}) \leq 3\tilde{q}\epsilon_1^{1/8} + 2^{-n}.$$

Proof. We know that $\Pr_{\mathsf{f}}\left[\Pr_x[(x,\mathsf{f}) \text{ is not good}] > \epsilon_1^{1/8}\right] \leq 3\tilde{q}\epsilon_1^{1/8}$. We say f is convenient if $\Pr_x[(x,f) \text{ is not good}] \leq \epsilon_1^{1/8}$. Fix a convenient f

$$\begin{aligned}&\Pr_R[(R,f) \notin \mathcal{G}] \\ &\leq \sum_m \prod_{i=1}^{l} \left(\Pr_{R_i}[(i, m \oplus R_i) \text{ is not good in } f]\right) \\ &\leq 2^n \times \left(\epsilon_1^{1/8}\right)^l \leq 1/2^n.\end{aligned}$$

In the first step, the sum is taken over $m \in \{0,1\}^n$. The last inequality follows from $l > n$, and $\epsilon_1 \leq \frac{1}{2^{16}}$. Hence, we have

$$\begin{aligned}\Pr_{R,\mathsf{f}}((R,\mathsf{f}) \notin \mathcal{G}) &\leq \Pr_{\mathsf{f}}[\mathsf{f} \text{ is not convenient}] + \Pr_{R}[(R,\mathsf{f}) \notin \mathcal{G} | \mathsf{f} \text{ is convenient}] \\ &\leq 3\tilde{q}\epsilon_1^{1/8} + 1/2^n. \qquad \square\end{aligned}$$

5 Crooked-Indifferentiability of Enveloped XOR construction

In this section we analyze the crooked-indifferentiability security of the EXor construction. Our main result in this section is Theorem 2.

Theorem 2. *Let $l = 3n+1, \tilde{q} \leq 2^{n/2}$ and $\epsilon_1 = \epsilon + \frac{q_1}{(l+1)2^n} \leq \frac{1}{16}$. Let $f : [l] \times \{0,1\}^n \to \{0,1\}^n$ be a family of random functions and $\mathsf{EXor} : \{0,1\}^n \to \{0,1\}^n$ be the enveloped-xor construction. Then there exists a simulator S such that for all all $((q_1,\tilde{q}),(q_2,q_{sim}),\epsilon,\delta)$ crooked distinguisher $\mathcal{A} = (\mathcal{A}_1,\mathcal{A}_2)$*

$$\mathbf{Adv}^{\mathsf{crooked\text{-}indiff}}_{\mathcal{A},(\mathsf{EXor},f)} \leq (4l^2\tilde{q})q_2^2/2^n + (4\tilde{q}+2l)q_2\epsilon_1^{1/16}$$

The simulator is described in Fig 2 which makes at most q_2 query to the random oracle $\mathcal{F}$ and makes $q_2 l\tilde{q}$ many calls to the subverted implementation $\tilde{f}$.

Proof. We recall that, in the real world, the distinguisher is interacting with the subverted construction $\widetilde{\mathsf{EXor}}$ which is defined as

$$\widetilde{\mathsf{EXor}}(R,m) = \tilde{f}(0,\tilde{g}_R(m)) \quad \text{where} \quad \tilde{g}_R(m) = \bigoplus_{i=1}^{l} \tilde{f}(i, m \oplus r_i).$$

We also define a hybrid construction $\overline{\mathsf{EXor}}[f](R,m) = f(0,\tilde{g}_R(m))$. Now consider an adversary $\mathcal{A}$ interacting with $(\mathsf{f}, \overline{\mathsf{EXor}} := \overline{\mathsf{EXor}}[\mathsf{f}])$ in the second phase.

Bad Events We consider the bad event happening immediately after ith query of the adversary which is of the form (j,x_i) for $j > 0$. We write $m_i = x_i + R_j$. We define four bad events.

1. Bad1$_i$ holds if $(0,\tilde{g}_R(m_i)) \in \mathsf{subv}_f$
2. Bad2a_i holds if $(0,\tilde{g}_R(m_i)) \in \mathsf{Dom}(\tau_{i-1})$
3. Bad2b_i holds if $\tilde{g}_R(m_i) = \tilde{g}(m_j)$ for some $j < i$
4. Bad2c_i holds if $(0,\tilde{g}_R(m_i)) \in Q(x)$ for some $x \in \mathsf{Dom}(\tau_i)$ and $x \in \mathsf{subv}_f$.

Let Bad1 $= \vee_i$Bad1$_i$, Bad2 $= \vee_i$(Bad2$a_i \vee$ Bad2$b_i \vee$ Bad2c_i), and Bad $=$ Bad1 $\vee$ Bad2.

Claim 3

$$\Delta_{\mathcal{A}_2(r,\tilde{\tau},R)}\big((f,\overline{\mathsf{EXor}}(R,\cdot)) \;;\; (f,\widetilde{\mathsf{EXor}}(R,\cdot))\big) \leq \Pr(\text{Bad1})$$

where Bad1 *holds while $\mathcal{A}$ interacting with* $(\mathsf{f},\overline{\mathsf{EXor}})$.

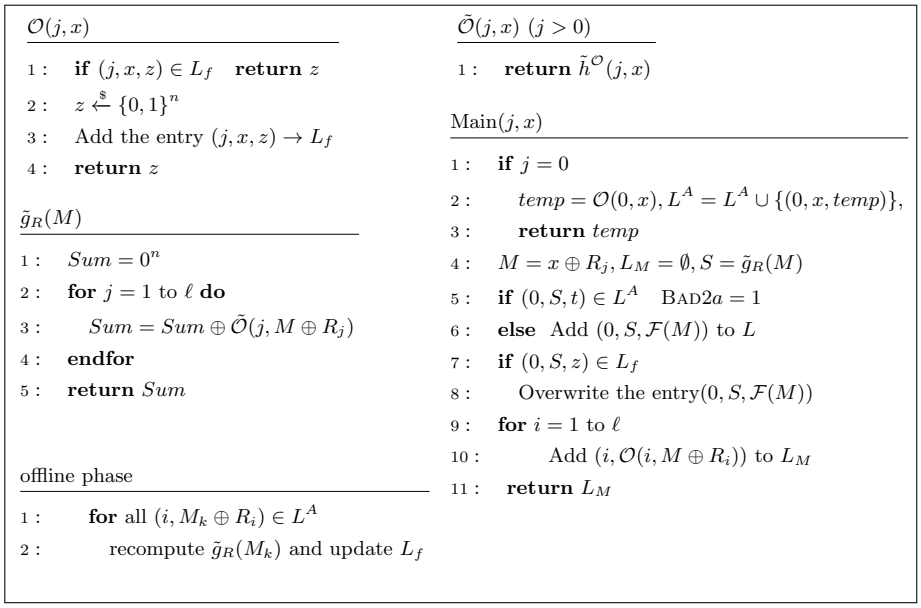

Fig. 2: Simulator for EXor: Offline Phase is executed after all the distinguisher queries.

Proof of the above claim is straightforward as both worlds behave identically until BAD1 does not hold.

We have defined our simulator $S^{\mathcal{F}}$ in Figure 2 where $\mathcal{F} : \{0,1\}^n \to \{0,1\}^n$ is a random function. The simulator has also observed the above bad events in particular, BAD2. Now we claim that the hybrid construction and the ideal world is indistinguishable provided BAD2 does not hold (in the hybrid world) while $\mathcal{A}$ interacting with $(\mathsf{f}, \overline{\mathsf{EXor}})$.

Claim 4

$$\Delta_{\mathcal{A}_2(r,\tilde{\tau},R)}\big((f, \overline{EXor}(R,\cdot)) \; ; \; (S^{\mathcal{F},\tilde{f}}(\tilde{\tau},R),\mathcal{F})\big) \leq \Pr(\text{BAD2}).$$

We call a transcript good if BAD2 does not hold. In case of simulator world, whenever BAD2 does not hold, simulator maintains extended transcript which is consistent with the hybrid world. As the simulator set all outputs of the function either randomly or through outputs of $\mathcal{F}$, realizing any such good transcript τ' has probability $2^{-n\sigma}$ where $\sigma = |\tau' \setminus \tau_0|$. We have already seen that the probability of realizing a good transcript in the hybrid world is exactly $2^{-n\sigma}$. In other words, the both worlds behave identically until BAD2 does not hold. Combining Claims 3 and 4, we get

$$\mathbf{Adv}^{\mathsf{crooked\text{-}indiff}}_{\mathcal{A},(\mathsf{EXor},f)} \leq \Pr[\,\text{BAD}\,].$$

The proof of Theorem 2 follows from the following lemma. □

Lemma 4.

$$\Pr[\textsc{Bad}] \leq (4l^2\tilde{q})q_2^2/2^n + (4\tilde{q} + 2l)q_2\epsilon_1^{1/16}$$

The lemma is proved in section 6.

6 Proof of Lemma 4

We write $f \Rightarrow_j \mathsf{Tr}_j$ to denote the event that after j queries to $(f, \overline{\mathsf{EXor}})$, an adversary obtains the transcript Tr_j. We skip the notation j if it is understood from the context.

Definition 7. *A transcript Tr_{i-1} is good if*

$$\Pr((R, \mathsf{f}) \in \mathcal{G} \mid \mathsf{f} \Rightarrow \mathsf{Tr}_{i-1}) \geq 1 - 3\tilde{q}\epsilon_1^{1/16}.$$

Applying Markov inequality on Lemma 3, we have $\Pr(\mathsf{Tr}_{i-1}$ is good$) \geq 1 - \epsilon_1^{1/16}$. Let us fix a good transcript Tr_{i-1} (which also determines m_i for the ith query) and a function f agreeing on Tr_{i-1} such that $(R, f) \in \mathcal{G}$.

Definition 8. *For any fix k, we say that f is called Tr_i-good if (i) $f \Rightarrow \mathsf{Tr}_{i-1}$ and (ii) (α_k, f) is good.*

Claim 5 *For any Tr_i-good f there exists a set S of size at least $2^n(1 - \epsilon_1^{1/4})$ such that for all $\beta \in S$, $f_\beta := f|_{\alpha \to \beta}$ is also Tr_i-good.*

Proof. We fix a function $f \in \Gamma_{R,\tau_{i-1},\pi_{i-1}}$ such that (α_k, f) is Tr_i-good. Now we identify a set of good values of β such that $f_\beta := f|_{\alpha_k \to \beta} \in \Gamma_{R,\tau_{i-1},\pi_{i-1}}$ such that (α_k, f) is Tr_i-good. In other words, setting the output of f on the point α_k to β keeps the pair (α_k, f_β) good. For every $x \in \mathsf{Dom}(\tau_{i-1}) \cap Q_f^{-1}(\alpha_k)$, let B_x denote the set of all bad β values for which good condition of (α_k, f) gets violated. By definition, $|B_x| \leq \epsilon_1^{1/2}$ and hence $|\cup_x B_x| \leq 2^n\epsilon_1^{1/4}$. We define

$$S = \mathcal{D} \setminus \cup_{x \in Q_f^{-1}(\alpha_k)} B_x.$$

Note that for all $\beta \in S$, (α_k, f_β) is Tr_i-good. □

Due to the above claim, we have

$$\Pr(\mathsf{f}(\alpha_k) = z \mid (\alpha_k, \mathsf{f}) \text{ is } \mathsf{Tr}_i\text{-good},) \leq \frac{1}{|S|} \leq \frac{1}{2^n(1 - \epsilon_1^{1/4})} \leq \frac{2}{2^n}.$$

The last inequality holds because $\epsilon_1 \leq \frac{1}{16}$. Now note that for any event E, we have

$$\begin{aligned}
\Pr(E|\mathsf{Tr}_{i-1}) &\leq \Pr_{\mathsf{f}}(E \wedge (R, \mathsf{f}) \in \mathcal{G}|\mathsf{Tr}_{i-1}) + 3\tilde{q}\epsilon_1^{1/16} \\
&\leq \sum_{k=1}^{l} \Pr_{\mathsf{f}}(E \wedge (\alpha_k, \mathsf{f}) \text{ is } \mathsf{Tr}_{i-1}\text{-good} \mid \mathsf{Tr}_{i-1}) + 3\tilde{q}\epsilon_1^{1/16} \\
&\leq \sum_{k=1}^{l} \Pr_{\mathsf{f}}(E \mid (\alpha_k, \mathsf{f}) \text{ is } \mathsf{Tr}_{i-1}\text{-good }) + 3\tilde{q}\epsilon_1^{1/16}
\end{aligned}$$

For the last inequality we simply use the fact that

$$\Pr_{\mathsf{f}}((\alpha_k, \mathsf{f}) \text{ is } \mathsf{Tr}_{i-1}\text{-good} \mid \mathsf{Tr}_{i-1}) \leq 1.$$

Now we bound individually each bad events and then we can multiply by l then add all the terms to get the bound.

Bound of $\mathbf{Pr}(\textsc{Bad}\mathbf{2}\boldsymbol{a_i} \cup \textsc{Bad}\mathbf{2}\boldsymbol{b_i})$ Fix a Tr_i-good f. Let $B2$ denote the set of all elements containing $\tilde{g}_R(m_j)$ (for all $j < i$) and all elements from $\mathcal{D}(\tau_{i-1})$ of the form $(0, *)$. Note that the set $B2$ and $\sum_{j \neq k} \tilde{f}(m_i + R_j)$ does not depend on the value $f(\alpha_k)$ provided $f(\alpha_k) \in S$. Hence,

$$\Pr_{\mathsf{f}}(\textsc{Bad}2a_i \cup \textsc{Bad}2b_i \mid (\alpha_k, \mathsf{f}) \text{ is } \mathsf{Tr}_{i-1}\text{-good }) \leq 2i/2^n.$$

Bound of $\mathbf{Pr}(\textsc{Bad}\mathbf{2}\boldsymbol{c_i})$ We first note that for all $\beta \in S$ and an input x which queries α_k, x is not crooked and a robust point. Let $A = \mathcal{D}(\tau_i) \backslash (\{\alpha_k\} \cup Q_f^{-1}(\alpha_k))$. Let $\tilde{A}$ denote the set of all points queried by the elements of A. Suppose $\tilde{g}_R(m_i) \notin \tilde{A}$. Then, for every x from the domain of τ_i querying $\tilde{g}_R(m_i)$ must query α_k and hence $\textsc{Bad}2c_i$ does not hold. So, $\textsc{Bad}2c_i$ can hold only if $\tilde{g}_R(m_i) \in \tilde{A}$. Once again by randomness of $f(\alpha_k)$, we have

$$\Pr(\textsc{Bad}2c_i \mid (\alpha_k, \mathsf{f}) \text{ is } \mathsf{Tr}_{i-1}\text{-good }) \leq 2\tilde{q}il/2^n.$$

Bound of $\mathbf{Pr}(\textsc{Bad}\mathbf{1}_\boldsymbol{i})$ Clearly, $\tilde{f}_\beta(x)$ can be different from $\tilde{f}(x)$, only if $x \in Q_f^{-1}(\alpha_k)$. Moreover for every $x \in Q_f^{-1}(\alpha_k)$, as both (α_k, f) and (α_k, f_β) are good, it holds that $x \notin \mathsf{subv}_f$ and $x \notin \mathsf{subv}_{f_b}$. Thus for any such Tr_i-good f, f_β, we have the following conditions: $\mathsf{subv}_f = \mathsf{subv}_{f_\beta}$. Thus,

$$\Pr[\textsc{Bad}1_i \mid (\alpha_k, f) \text{ good}, \mathsf{Tr}_i \text{ good}] \leq 2\epsilon_1$$

So,

$$\Pr[\textsc{Bad}_i \mid \mathsf{Tr}_{i-1}] \leq 4l^2 q_2 \tilde{q}/2^n + 2l\epsilon_1 + 3\tilde{q}\epsilon_1^{1/16}$$

Finally, we add the probability that we realize a not good transcript Tr_{i-1} and we obtain bound for $\Pr(\textsc{Bad}_i)$. By taking union bound over $i \in [q_2]$, we get

$$\begin{aligned}\Pr[\textsc{Bad}] &\leq 4l^2 q_2^2 \tilde{q}/2^n + 2lq_2\epsilon_1 + 3\tilde{q}q_2\epsilon_1^{1/16} + q_2\epsilon_1^{1/16} \\ &\leq (4l^2\tilde{q})q_2^2/2^n + (4\tilde{q} + 2l)q_2\epsilon_1^{1/16}\end{aligned}$$

This finishes the proof of Lemma 4. □

References

1. Andreeva, E., Mennink, B., Preneel, B.: On the indifferentiability of the Grøstl hash function. In: Garay, J.A., Prisco, R.D. (eds.) SCN 10. LNCS, vol. 6280, pp. 88–105. Springer, Heidelberg (Sep 2010). https://doi.org/10.1007/978-3-642-15317-4_7 1
2. Bellare, M., Paterson, K.G., Rogaway, P.: Security of symmetric encryption against mass surveillance. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 1–19. Springer, Heidelberg (Aug 2014). https://doi.org/10.1007/978-3-662-44371-2_1 1
3. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (May / Jun 2006). https://doi.org/10.1007/11761679_25
4. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the indifferentiability of the sponge construction. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 181–197. Springer, Heidelberg (Apr 2008). https://doi.org/10.1007/978-3-540-78967-3_11 1
5. Bhattacharyya, R., Mandal, A., Nandi, M.: Indifferentiability characterization of hash functions and optimal bounds of popular domain extensions. In: Progress in Cryptology - INDOCRYPT 2009, 10th International Conference on Cryptology in India, New Delhi, India, December 13-16, 2009. Proceedings. pp. 199–218 (2009), `https://doi.org/10.1007/978-3-642-10628-6_14` 1, 1.2, 1.2
6. Bhattacharyya, R., Mandal, A., Nandi, M.: Security analysis of the mode of JH hash function. In: Hong, S., Iwata, T. (eds.) Fast Software Encryption, 17th International Workshop, FSE 2010,. Lecture Notes in Computer Science, vol. 6147, pp. 168–191. Springer (2010) 1
7. Chang, D., Nandi, M.: Improved indifferentiability security analysis of chopMD hash function. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 429–443. Springer, Heidelberg (Feb 2008). https://doi.org/10.1007/978-3-540-71039-4_27 1
8. Coretti, S., Dodis, Y., Guo, S., Steinberger, J.P.: Random oracles and non-uniformity. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 227–258. Springer, Heidelberg (Apr / May 2018). https://doi.org/10.1007/978-3-319-78381-9_9 1
9. Coron, J.S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård revisited: How to construct a hash function. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 430–448. Springer, Heidelberg (Aug 2005). https://doi.org/10.1007/11535218_26 1
10. Degabriele, J.P., Paterson, K.G., Schuldt, J.C.N., Woodage, J.: Backdoors in pseudorandom number generators: Possibility and impossibility results. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 403–432. Springer, Heidelberg (Aug 2016). https://doi.org/10.1007/978-3-662-53018-4_15 1
11. Dodis, Y., Guo, S., Katz, J.: Fixing cracks in the concrete: Random oracles with auxiliary input, revisited. In: Coron, J.S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part II. LNCS, vol. 10211, pp. 473–495. Springer, Heidelberg (Apr / May 2017). https://doi.org/10.1007/978-3-319-56614-6_16 1
12. Dodis, Y., Reyzin, L., Rivest, R.L., Shen, E.: Indifferentiability of permutation-based compression functions and tree-based modes of operation, with applications to MD6. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 104–121. Springer, Heidelberg (Feb 2009). https://doi.org/10.1007/978-3-642-03317-9_7 1

13. Maurer, U.M., Renner, R., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer, Heidelberg (Feb 2004). https://doi.org/10.1007/978-3-540-24638-1_2 1
14. Mennink, B.: Indifferentiability of double length compression functions. In: Stam, M. (ed.) 14th IMA International Conference on Cryptography and Coding. LNCS, vol. 8308, pp. 232–251. Springer, Heidelberg (Dec 2013). https://doi.org/10.1007/978-3-642-45239-0_14 1
15. Moody, D., Paul, S., Smith-Tone, D.: Improved indifferentiability security bound for the JH mode. Cryptology ePrint Archive, Report 2012/278 (2012), `http://eprint.iacr.org/2012/278` 1
16. Naito, Y.: Indifferentiability of double-block-length hash function without feed-forward operations. In: Pieprzyk, J., Suriadi, S. (eds.) ACISP 17, Part II. LNCS, vol. 10343, pp. 38–57. Springer, Heidelberg (Jul 2017) 1
17. Russell, A., Tang, Q., Yung, M., Zhou, H.S.: Cliptography: Clipping the power of kleptographic attacks. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 34–64. Springer, Heidelberg (Dec 2016). https://doi.org/10.1007/978-3-662-53890-6_2 1
18. Russell, A., Tang, Q., Yung, M., Zhou, H.S.: Generic semantic security against a kleptographic adversary. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017. pp. 907–922. ACM Press (Oct / Nov 2017). https://doi.org/10.1145/3133956.3133993 1
19. Russell, A., Tang, Q., Yung, M., Zhou, H.S.: Correcting subverted random oracles. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 241–271. Springer, Heidelberg (Aug 2018). https://doi.org/10.1007/978-3-319-96881-0_9 1, 1.1, 1.2, 2.1, 4, 3, 3, 3.1, 1, 3.2
20. Russell, A., Tang, Q., Yung, M., Zhou, H., Zhu, J.: Correcting subverted random oracles. IACR Cryptol. ePrint Arch. **2021**, 42 (2021), `https://eprint.iacr.org/2021/042` 1.2
21. Young, A., Yung, M.: The dark side of "black-box" cryptography, or: Should we trust capstone? In: Koblitz, N. (ed.) CRYPTO'96. LNCS, vol. 1109, pp. 89–103. Springer, Heidelberg (Aug 1996). https://doi.org/10.1007/3-540-68697-5_8 1