Indifferentiability of \textbf{SKINNY-HASH}
Internal Functions

Akinori Hosoyamada\textsuperscript{1,2} and Tetsu Iwata\textsuperscript{2}

\textsuperscript{1}NTT Secure Platform Laboratories, NTT Corporation, Tokyo, Japan
akinori.hosoyamada.bh@hco.ntt.co.jp

\textsuperscript{2}Department of Information and Communication Engineering, Nagoya University, Nagoya, Japan
{hosoyamada.akinori,tetsu.iwata}@nagoya-u.jp

Abstract

We provide a formal proof for the indifferentiability of SKINNY-HASH internal function from a random oracle. SKINNY-HASH is a family of function-based sponge hash functions, and it was selected as one of the second round candidates of the NIST lightweight cryptography competition. Its internal function is constructed from the tweakable block cipher SKINNY. The construction of the internal function is very simple and the designers claim \(n\)-bit security, where \(n\) is the block length of SKINNY. However, a formal security proof of this claim is not given in the original specification of SKINNY-HASH. In this paper, we formally prove that the internal function of SKINNY-HASH has \(n\)-bit security, i.e., it is indifferentiable from a random oracle up to \(O(2^n)\) queries, substantiating the security claim of the designers.

Keywords: symmetric-key cryptography, provable security, sponge construction, indifferentiability, SKINNY, SKINNY-HASH

1 Introduction

The sponge construction is one of the most basic constructions to convert a function or permutation into a cryptographic hash function. It is used in many modern cryptographic hash functions including SHA-3 \cite{Nat15}.

The sponge construction based on \(F : \{0,1\}^b \rightarrow \{0,1\}^b\), where \(F\) is a public permutation or a public function, has two positive parameters \(r\) and \(c\) such that \(r + c = b\). Given an input \(M \in \{0,1\}^*\), the hash value is computed as follows: First, \(M\) is padded so that its length is a multiple of \(r\). Let \(M[1]||\cdots||M[L] \in \{0,1\}^{rL}\) be the message after padding, where \(M[i] \in \{0,1\}^r\) for each \(i\). Second, the internal states \(s_0, \ldots, s_L \in \{0,1\}^b\) are computed in a sequential order as \(s_0 := IV\) and \(s_i := F(s_{i-1} \oplus (M[i]||0^c))\) for \(1 \leq i \leq L\), where \(IV \in \{0,1\}^b\) is an initialization vector. (This phase is called the absorbing phase.) Third, the
internal states $st_{L+1}, \ldots, st_{L+h-1}$ and the output value $H = H[1] \cdots || H[h] \in \{0, 1\}^h$ ($H[i] \in \{0, 1\}^r$) are computed as $st_{L+i} := F(st_{L+i-1})$ for $1 \leq i \leq h - 1$ and $H[i] := \text{(the most significant } r \text{ bits of } st_{L+i-1})$. (This phase is called the squeezing phase.) $H$ is truncated if necessary. See Fig. 1.

![Sponge Construction Diagram](attachment:image.png)

Figure 1: The sponge construction.

The sponge construction is proven to be indifferentiable from a random oracle up to $O(2^{c/2})$ queries when $F$ is a random oracle or an ideal permutation [BDPA08], and an appropriate padding function is chosen. That is, if a cryptosystem is proven to be secure in the random oracle model, the security of the cryptosystem does not decrease even if we replace the random oracle with the sponge construction, as long as the number of queries made to $F$ through the sponge construction or the direct computation of $F$ (and $F^{-1}$, if $F$ is a permutation) is $O(2^{c/2})$.

Since the sponge construction is proven to be secure, to realize a secure cryptographic hash function, it is sufficient to construct a secure function or permutation $F$. There are two possible ways to realize such $F$.

One approach is to design a dedicated function or permutation from scratch. Most sponge-based hash functions including SHA-3 take this approach. For instance, SHA-3 uses a dedicated 1600-bit permutation as $F$. The other approach is to construct $F$ from well-established primitives such as block ciphers or tweakable block ciphers, which is taken by the SKINNY-HASH function family.

---

1In some concrete hash functions, the parameters $r$ and $c$ are changed to other parameters $r'$ and $c'$ such that $r' + c' = b$ in the squeezing phase.
1.1 SKINNY-HASH Internal Functions

SKINNY-HASH [BJK+20] is a family of function-based sponge constructions, which is one of the second round candidates of the NIST lightweight cryptography competition [NIS]. It consists of SKINNY-tk2-Hash and SKINNY-tk3-Hash, which are the sponge constructions with $b = 256$ and $b = 384$, and the internal functions are built with the tweakable block ciphers SKINNY-128-256 and SKINNY-128-384 [BJK+16], respectively.

SKINNY-128-256 is a tweakable permutation $\tilde{E}^{256}_{tk}: \{0,1\}^{128} \rightarrow \{0,1\}^{128}$, where the tweakey $tk$ is chosen from $\{0,1\}^{256}$. Similarly, SKINNY-128-384 is a tweakable permutation $\tilde{E}^{384}_{tk}$ on $\{0,1\}^{128}$, where the tweakey $tk$ is chosen from $\{0,1\}^{384}$. $\tilde{E}^{256}_{tk}$ and $\tilde{E}^{384}_{tk}$ are expected to be secure and suitable to instantiate ideal ciphers of which the block length is 128 bits and the key lengths are 256 bits and 384 bits, respectively.

The internal functions $F_{256}: \{0,1\}^{256} \rightarrow \{0,1\}^{256}$ and $F_{384}: \{0,1\}^{384} \rightarrow \{0,1\}^{384}$ of SKINNY-tk2-Hash and SKINNY-tk3-Hash are defined by

$$F_{256}(x) := \tilde{E}^{256}_{x}(c_1) || \tilde{E}^{256}_{x}(c_2)$$

and

$$F_{384}(x) := \tilde{E}^{384}_{x}(c_1) || \tilde{E}^{384}_{x}(c_2) || \tilde{E}^{384}_{x}(c_3),$$

respectively, where $c_1, c_2, c_3$ are distinct 128-bit constants (see Fig. 2).

Figure 2: The SKINNY-HASH internal functions $F_{256}$ and $F_{384}$.

In the specification of SKINNY-HASH, the designers claim that “The function $F_{256}$ is indifferentiable from a 256-bit random function up to $O(2^{128})$ queries.” and “The same intuitive argument applies to $F_{384}$. However, the bound is worse than the one for $F_{256}$ by a factor of 3...”.
Their design and security claim are notable since \( F_{256} \) and \( F_{384} \) achieve \( n \)-bit security from an \( n \)-bit tweakable block cipher although the designs of the functions are quite simple (just a few parallel applications of tweakable block ciphers). On the other hand, when we build a compression function (to be used in the Merkle-Damgård construction) based on (tweakable) block ciphers, even the known approaches to achieve the same level of security require more complex constructions [Nai11, HK14].

Observe that \( F_{256} \) and \( F_{384} \) do not give a perfect random function. If we write \( F_{256}(x) = Y_1||Y_2 \), then \( Y_1 = Y_2 \) never happens. Similarly, if we write \( F_{384}(x) = Y_1||Y_2||Y_3 \), then for any \( i \neq j \), \( Y_i = Y_j \) is impossible. The \( n \)-bit security claim comes from the intuition that these are the only events that make them different from a truly random function. However, there is no formal proof for the \( n \)-bit security claim. Generally, it is highly favorable that a mode of operation of (tweakable) block ciphers has formal security proofs when a security claim is provided.

1.2 Our Contributions

In this paper, we give a formal proof of the indifferentiability of the SKINNY-HASH internal functions \( F_{256} \) and \( F_{384} \) in the ideal cipher model. In fact, we show a more general theorem: Let \( E \) be an \( n \)-bit block cipher with \( \ell n \)-bit key, where \( \ell \) is a small constant. Define \( F^E : \{0,1\}^{\ell n} \rightarrow \{0,1\}^{\ell n} \) be the function defined by

\[
F^E(x) := E_x(c_1)|| \cdots ||E_x(c_\ell),
\]

where \( c_1, \ldots, c_\ell \) are fixed distinct \( n \)-bit constants. We call \( F^E \) the SHI function ("SHI function" is an abbreviation of SKINNY-HASH Internal function). We show the following theorem.

**Theorem 1** (Main theorem, informal). If \( E \) is an ideal cipher, the SHI function \( F^E \) is indifferentiable from a random oracle as long as the total number of queries made to \( E \) and its inverse \( E^{-1} \) are in \( o(2^n) \).

This theorem shows that the SHI function has \( n \)-bit security, as claimed by the designers. Since the structure of SKINNY-HASH internal functions and the generalization \( F^E \) is quite simple and the security is very high, we believe that more and more function-based sponge constructions will be developed and used in practical situations relying on the SHI construction and our security proof.

**Intuition of the proof for Theorem 1**

Intuitively, we construct a simulator \( S \) as follows:

When an adversary \( A \) queries a value \((K,X)\) to \( E \) that \( A \) has already queried before, \( S \) just returns the previous result stored in a list \( L_K \).

---

2Our intuition for the simulator is based on “Rationale of \( F_{256} \) and \( F_{384} \)” in the original specification [BJK+20]. Note that the original explanation in [BJK+20] is very rough (only two paragraphs) and it is not trivial how to derive a formal security proof from that.
When $A$ queries a fresh value $(K, X)$ to $E$ such that $A$ has never queried $(K, Z)$ for any $Z$ to $E$ nor $E^{-1}$, $S$ first queries $K$ to the random oracle $RO : \{0, 1\}^{n\ell} \to \{0, 1\}^{n\ell}$, and simulates the values $E_K(c_1), \ldots, E_K(c_i)$ as $E_K(c_1)|\cdots|E_K(c_i) := RO(K)$. $S$ stores the pairs $(c_1, E_K(c_1)), \ldots, (c_i, E_K(c_i))$ into $L_K$. If $X = c_i$ for some $i$, then $S$ returns the value $E_K(c_i)$ to $A$. If $X \neq c_i$ for all $i$, then $S$ picks a value $Y$ from $\{0,1\}^n \setminus \{E_K(c_1), \ldots, E_K(c_i)\}$ uniformly at random, simulates the value $E_K(X)$ as $E_K(X) := Y$, stores the pair $(X, Y)$ into the list $L_K$, and returns $Y$ to $A$.

When $A$ queries a value $(K, X)$ to $E$ such that $A$ has already queried $(K, Z)$ for some $Z$ to $E$ or $E^{-1}$ before but $(X, Y) \notin L_K$ for any $Y$, $S$ chooses $Y$ from $\{0,1\}^{n\ell} \to \{0,1\}^{n\ell}$ randomly in such a way that $Y \neq Y'$ holds for every pair $(X', Y') \in L_K$, stores the pair $(X, Y)$ into the list $L_K$, and returns $Y$ to $A$.

Queries to $E^{-1}$ are simulated in the same way.

The above simulation fails only when $S$ queries $K$ to the random oracle $RO$, and $RO(K) = Y_1|\cdots|Y_i \ (Y_i \in \{0,1\}^n \text{ for each } i)$ happens to satisfy $Y_i = Y_j$ for some $i \neq j$. Roughly speaking, the probability of this event can be upper bounded by $O(1/2^n)$ for each $K$, and thus the failure probability of $S$ is always negligibly small if the number of queries made by $A$ is smaller than $2^n$. Note that such an event never holds in the real world since, if we divide $E^E(K) \in \{0, 1\}^{n\ell}$ into $n$-bit blocks as $E^E(K) = Y_1|\cdots|Y_{\ell}$, then $Y_i = E_K(c_i)$ never matches $Y_j = E_K(c_j)$ for $i \neq j$, for arbitrary $K$.

The main contribution of the paper is to provide a formal proof that the above intuition is correct.

### 1.3 Related Work

The SHI function is quite similar to a function proposed in a previous work [CNL+08, Section 4.4]. The difference of the SHI function from the function in [CNL+08] is that, while the domain and the range of the SHI function are the same since it is supposed to be used in the sponge construction, the domain of the function in [CNL+08] is larger than its range since it is supposed to be used as a compression function in the Merkle-Damgård construction. In addition, while the previous work shows collision-resistance, this paper shows the indifferiability.

### 1.4 Paper Organization

Section 2 gives basic notations and definitions. Section 3 shows a formal proof for the SKINNY-HASH internal function. Section 4 concludes the paper.

### 2 Preliminaries

We say that a function $f : \mathbb{Z}_{\geq 0} \to \mathbb{R}$ is negligible if, for arbitrary constant $c > 0$, there exists a sufficiently large integer $N$ such that $|f(n)| \leq 1/n^c$ for all $n \geq N$. 

5
**Block ciphers.** An $n$-bit block cipher with $k$-bit keys is a keyed permutation $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$. In other words, the function $E$ is called a block cipher when $E(K, \cdot) : \{0,1\}^n \rightarrow \{0,1\}^n$ is a permutation for all $K \in \{0,1\}^k$. Let $E^{-1}$ denote the inverse of $E$ defined by $E^{-1}(K, E(K, M)) = M$ for all $M \in \{0,1\}^n$. We often write $E_K(\cdot)$ and $E^{-1}_K(\cdot)$ instead of $E(K, \cdot)$ and $E^{-1}(K, \cdot)$, respectively.

**Ideal primitive models.** The random oracle model is the model in which there exists the oracle of a random function $RO$ (either of fixed input-length and variable input-lengths), and adversaries have access to $RO(\cdot)$. The ideal permutation model is the model in which there exists the oracle of a random permutation $P$ and its inverse $P^{-1}$, and adversaries have access to $P(\cdot)$ and $P^{-1}(\cdot)$ (we sometimes refer to $P$ as an ideal permutation). The ideal cipher model is the model in which there exists the oracle of an ideal cipher $E$ (an ideal cipher is a block cipher such that, for each key $K$, $E(K, \cdot)$ is chosen independently at random) and its inverse $E^{-1}$, and adversaries have access to $E(\cdot, \cdot)$ and $E^{-1}(\cdot, \cdot)$. In what follows, we refer to (i) a random oracle (either of fixed input length and variable input lengths), (ii) an ideal permutation, and (iii) an ideal cipher as **ideal primitives**.

**Indifferentiability.** Let $\mathcal{R}$ be an ideal primitive. Let $H$ be a function that accesses to the oracle of another ideal primitive $O$, and suppose that the input and output lengths of $H$ are the same as those of $\mathcal{R}$. Let $S$ be an algorithm that has the same interface of input and output as $O$ and has an oracle access to $\mathcal{R}$. Let $\text{Real}^{H,O,A}$ be the game that runs $A$ relative to $(H^O,O)$, and finally returns what $A^{H^O,O}$ outputs. In addition, let $\text{Ideal}^{R,S,A}_S$ be the game that runs $A$ relative to $(\mathcal{R},S^R)$, and finally returns what $A^{R,S^R}$ outputs. We define the indifferentiability advantage of an adversary $A$ against $(H^O,O)$ and $\mathcal{R}$ with respect to the simulator $S$ by

$$\text{Adv}^{\text{indiff}}_{(H^O,O),\mathcal{R},S}(A) := \Pr\left[1 \leftarrow \text{Real}^{H,O,A}\right] - \Pr\left[1 \leftarrow \text{Ideal}^{R,S,A}_S\right].$$

See also Fig. 3.

**Definition 1 (Indifferentiability [MR04]).** The function $H^O$ is said to be $(t_S,t_A,q_A,Q_A,\epsilon)$-indifferentiable from $\mathcal{R}$ if there exists a simulator $S$ such that (1) $S$ runs in time at most $t_S$, and (2) for any adversary $A$ that runs in time $t_A$, makes at most $q_A$ queries to $O$ (resp., $S^R$), and $Q_A$ queries to $H^O$ (resp., $\mathcal{R}$),

$$\text{Adv}^{\text{indiff}}_{(H^O,O),\mathcal{R},S}(A) \leq \epsilon$$

holds.

We ambiguously say that $H^O$ is indifferentiable from $\mathcal{R}$ up to $x$ queries if there exists a simulator $S$ such that, for arbitrary adversary $A$ such that $q_A, Q_A \ll x$, $\text{Adv}^{\text{indiff}}_{(H^O,O),\mathcal{R},S}(A)$ is negligible.

The following composition theorem assures that, if (i) the security of a primitive $Q$ is defined with a single-stage game, and (ii) $H^O$ is indifferentiable from
Theorem 2 (Composition theorem \[\text{RSS11}\]). Let $G$ be a single-stage game. Let $H$ and $\mathcal{O}$ as above. Then, for any adversary $A$ and simulator $S$, there exist adversaries $B$ and $C$ such that

$$\Pr[G^H,\mathcal{O} \Rightarrow x] \leq \Pr[G^R \Rightarrow x] + \text{Adv}^{\text{indiff}}_{R,O}(R,S(C))$$

and $Q_B \leq Q_A + Q_S \cdot q_A$, $Q_C \leq Q_G + n_{G,A} \cdot q_A$, $q_C \leq n_{G,A} \cdot q_A$, $t_B \leq t_A + q_A \cdot \tilde{t}_S$, $t_C \leq t_G + n_{G,A} \cdot t_A$ hold.\(^3\) Here, $Q_X$ denotes the maximum number of queries to $R$ or $H^\mathcal{O}$ made by $X$ for $X = A, B, C, G$, and $Q_S$ denotes the maximum number of queries made to $R$ at each invocation of $S$. $q_X$ denotes the maximum number of queries to $O$ or $S^R$ made by $X$ for $X = A, C$. $t_X$ denotes the maximum running times of $X$ for $X = A, B, C, G$, and $\tilde{t}_S$ denotes the maximum time that $S$ spends at each invocation of $S$. $n_{G,A}$ denotes the number of invocations of $A$ by $G$.

3 Security Proofs of the SHI Function

Let $E$ denote an $n$-bit ideal cipher with $\ell n$-bit keys, where $\ell$ is a small constant. Let $F^E$ be the SHI function defined as in \([1]\). The goal of this section is to prove the following theorem, which shows that the SHI function is indistinguishable from a random oracle up to $O(2^n)$ queries. Together with Theorem 2, the following theorem assures that the security of the sponge construction does not decrease when its internal function is instantiated with the SHI function up to $O(2^n)$ queries.

\(^3\)The claim on the number of queries and running times (the inequalities on $Q_B$, $Q_C$, $q_C$, $t_B$, and $t_C$) are a little bit different from the original statement in \[\text{RSS11}\], but they can be deduced from the arguments in the original proof.
Theorem 3. There exists a simulator $S$ that satisfies the following conditions.

1. $S$ makes at most 1 query to RO and returns an output in time $O(1)$ at each invocation of $S$.

2. For an arbitrary adversary $A$ that makes at most $Q_A$ queries to $H^E$ and makes $q_A$ queries to $E$ and $E^{-1}$ in total,

$$\text{Adv}^{\text{indiff}}_{(F^E,(E,E^{-1})),\text{RO},S}(A) \leq \frac{\ell^2(q_A + \ell Q_A)}{2^n}$$

holds.

Proof. We show the theorem with the code-based game-playing proof technique [BR06], by introducing 6 games $G_1, \ldots, G_6$. See the explanation below Theorem 1 for intuition of the proof.

**Game $G_1$.** $G_1$ is the real game, where the adversary $A$ runs relative to the oracles $F^E$, $E$, and $E^{-1}$. We assume that the oracle of the ideal cipher $E$ is implemented by using lazy sampling. See Fig. 4 for details.

**Games $G_2$ and $G_3$.** $G_2$ is identical to $G_1$ except that, when a value $(K,X)$ (resp., $(K,Y)$) is queried to $E$ (resp., $E^{-1}$) such that $(K,Z)$ has not been queried to $E$ nor $E^{-1}$ for any $Z$, the values $E_K(c_1), \ldots, E_K(c_\ell)$ are sampled before answering to the query. In addition, the sampling of $E_K(c_1), \ldots, E_K(c_\ell)$ are performed as follows:

1. Choose $Y_1, \ldots, Y_\ell \in \{0,1\}^n$ independently and uniformly at random.

2. If $Y_i = Y_j$ holds for some $i \neq j$, set flag to be bad, and re-sample $Y_1, \ldots, Y_\ell$ in such a way that $Y_i \neq Y_j$ holds for all $i \neq j$.

3. Set $E_K(c_i) := Y_i$ for $i = 1, \ldots, \ell$.

The procedure $F^E$ is not changed from $G_1$. $G_3$ is identical to $G_2$ except that the re-sampling of $Y_1, \ldots, Y_\ell$ is not performed even if flag is set to be bad. See Fig. 5 for details.

**Games $G_4$ and $G_5$.** In the game $G_4$, compared to $G_3$, a random oracle RO is introduced, and the sampling of $Y_1, \ldots, Y_\ell$ in $E$ and $E^{-1}$ when $L_K$ is empty is replaced with the query of $K$ to the random oracle RO. $F^E$ is not changed in $G_4$. The game $G_5$ is identical to $G_4$ except that $F^E$ is modified in such a way that $F^E(T) := \text{RO}(T)$. See Fig. 6 for details.

**Game $G_6$.** $G_6$ is the ideal game. In $G_6$, $A$ runs relative to RO and $S^{\text{RO}}$ instead of $F^E$ and $(E,E^{-1})$, where $S$ is a simulator defined as in Fig. 7. Given an input $(b,K,Z) \in \{0,1\} \times \{0,1\}^n \times \{0,1\}^n$, $S$ simulates $E(K,Z)$ if $b = 0$ and $E^{-1}(K,Z)$ if $b = 1$. The behavior of $S$ is the same as that of $E$ and $E^{-1}$ in the games $G_4$ and $G_5$.
Game $G_1^A$
\[ x \leftarrow A^{F_E,E,E^{-1}} \]
return $x$

Procedure $E(K,X)$
if there exists $Y$ such that $(X,Y) \in L_K$
return $Y$
e else
\[ Y \leftarrow \{0,1\} \setminus L_{K,\text{out}} \]
\[ L_{K,\text{in}} \leftarrow L_{K,\text{in}} \cup \{X\} \]
\[ L_{K,\text{out}} \leftarrow L_{K,\text{out}} \cup \{Y\} \]
\[ L_K \leftarrow L_K \cup \{(X,Y)\} \]

Procedure $E^{-1}(K,Y)$
if there exists $X$ such that $(X,Y) \in L_K$
return $X$
e else
\[ X \leftarrow \{0,1\} \setminus L_{K,\text{in}} \]
\[ L_{K,\text{in}} \leftarrow L_{K,\text{in}} \cup \{X\} \]
\[ L_{K,\text{out}} \leftarrow L_{K,\text{out}} \cup \{Y\} \]
\[ L_K \leftarrow L_K \cup \{(X,Y)\} \]

Procedure $F_E(T)$
\[ S \leftarrow E(T,c_1) \| \ldots \| E(T,c_{\ell}) \]
return $S$

Figure 4: The real game $G_1$. The lists $L_K$, $L_{K,\text{in}}$, and $L_{K,\text{out}}$ (for $K \in \{0,1\}^{n\ell}$) are set to be empty at the beginning of the game.

Below we give an upper bound of the indifferiability advantage $\text{Adv}^{\text{indiff}}_{(F_E,E,E^{-1}),\text{RO},S}(A)$. First, by definition of the games,

\[ |Pr[1 \leftarrow G_i^A] - Pr[1 \leftarrow G_i^{A+1}]| = 0 \]  \hspace{1cm} (2)

holds for $i = 1, 3, 4, 5$.

On the difference between $G_2$ and $G_3$, let $\text{SetBad}(i)$ denote the event that $\text{flag}$ is set to be $\text{bad}$ at the $i$-th query to $E$ or $E^{-1}$ (note that $1 \leq i \leq q_A + \ell \cdot Q_A$).
Procedure $E(K, X)$
if $L_K$ is empty
  $Y_1, \ldots, Y_\ell \leftarrow \{0, 1\}^n$
  if $Y_i = Y_j$ for some $i \neq j$
    flag $\leftarrow$ bad
  for $i = 1, \ldots, \ell$ do:
    $Y_i \leftarrow \{0, 1\}^n \setminus \{Y_1, \ldots, Y_{i-1}\}$
  $L_{K,\text{in}} \leftarrow L_{K,\text{in}} \cup \{c_1, \ldots, c_\ell\}$
  $L_{K,\text{out}} \leftarrow L_{K,\text{out}} \cup \{Y_1, \ldots, Y_\ell\}$
  $L_K \leftarrow L_K \cup \{(c_1, Y_1), \ldots, (c_\ell, Y_\ell)\}$
else if there exists $Y$ such that $(X, Y) \in L_K$
  return $Y$
else
  $Y \leftarrow \{0, 1\}^n \setminus L_{K,\text{out}}$
  $L_{K,\text{in}} \leftarrow L_{K,\text{in}} \cup \{X\}$
  $L_{K,\text{out}} \leftarrow L_{K,\text{out}} \cup \{Y\}$
  $L_K \leftarrow L_K \cup \{(X, Y)\}$

Procedure $E^{-1}(K, Y)$
if $L_K$ is empty
  $Y_1, \ldots, Y_\ell \leftarrow \{0, 1\}^n$
  if $Y_i = Y_j$ for some $i \neq j$
    flag $\leftarrow$ bad
  for $i = 1, \ldots, \ell$ do:
    $Y_i \leftarrow \{0, 1\}^n \setminus \{Y_1, \ldots, Y_{i-1}\}$
  $L_{K,\text{in}} \leftarrow L_{K,\text{in}} \cup \{c_1, \ldots, c_\ell\}$
  $L_{K,\text{out}} \leftarrow L_{K,\text{out}} \cup \{Y_1, \ldots, Y_\ell\}$
  $L_K \leftarrow L_K \cup \{(c_1, Y_1), \ldots, (c_\ell, Y_\ell)\}$
else if there exists $X$ such that $(X, Y) \in L_K$
  return $X$
else
  $X \leftarrow \{0, 1\}^n \setminus L_{K,\text{in}}$
  $L_{K,\text{in}} \leftarrow L_{K,\text{in}} \cup \{X\}$
  $L_{K,\text{out}} \leftarrow L_{K,\text{out}} \cup \{Y\}$
  $L_K \leftarrow L_K \cup \{(X, Y)\}$

Figure 5: The modified versions of $E(K, X)$ and $E^{-1}(K, Y)$ in the games $G_2$ and $G_3$. The steps surrounded by a square is performed in $G_3$ but not performed in $G_2$. 
Procedure $\text{RO}(T)$
if there exists $W$ s.t. $(T, W) \in L_{\text{RO}}$
return $W$
else
$W \leftarrow \{0, 1\}^{\alpha}$
$L_{\text{RO}} \leftarrow L_{\text{RO}} \cup \{(T, W)\}$
return $W$

Procedure $E(K, X)$
if $L_K$ is empty
$Y_1 || \cdots || Y_\ell \leftarrow \text{RO}(K)$ (here, $Y_i \in \{0, 1\}^n$ for each $i$)
$L_{K,\text{in}} \leftarrow L_{K,\text{in}} \cup \{c_1, \ldots, c_\ell\}$
$L_{K,\text{out}} \leftarrow L_{K,\text{out}} \cup \{Y_1, \ldots, Y_\ell\}$
$L_K \leftarrow L_K \cup \{(c_1, Y_1), \ldots, (c_\ell, Y_\ell)\}$
else if there exists $Y$ such that $(X, Y) \in L_K$
return $Y$
else
$Y \leftarrow \{0, 1\}^n \setminus L_{K,\text{out}}$
$L_{K,\text{in}} \leftarrow L_{K,\text{in}} \cup \{X\}$
$L_{K,\text{out}} \leftarrow L_{K,\text{out}} \cup \{Y\}$
$L_K \leftarrow L_K \cup \{(X, Y)\}$

Procedure $E^{-1}(K, Y)$
if $L_K$ is empty
$Y_1 || \cdots || Y_\ell \leftarrow \text{RO}(K)$ (here, $Y_i \in \{0, 1\}^n$ for each $i$)
$L_{K,\text{in}} \leftarrow L_{K,\text{in}} \cup \{c_1, \ldots, c_\ell\}$
$L_{K,\text{out}} \leftarrow L_{K,\text{out}} \cup \{Y_1, \ldots, Y_\ell\}$
$L_K \leftarrow L_K \cup \{(c_1, Y_1), \ldots, (c_\ell, Y_\ell)\}$
else if there exists $X$ such that $(X, Y) \in L_K$
return $X$
else
$X \leftarrow \{0, 1\}^n \setminus L_{K,\text{in}}$
$L_{K,\text{in}} \leftarrow L_{K,\text{in}} \cup \{X\}$
$L_{K,\text{out}} \leftarrow L_{K,\text{out}} \cup \{Y\}$
$L_K \leftarrow L_K \cup \{(X, Y)\}$

Procedure $F^E(T)$
$S \leftarrow E(T, c_1) || \cdots || E(T, c_\ell)$
$S \leftarrow \text{RO}(T)$
return $S$

Figure 6: The procedure $\text{RO}$ and the modified versions of $E(K, X)$, $E^{-1}(K, Y)$, and $F^E$ in the games $G_4$ and $G_5$. The list $L_{\text{RO}}$ is set to be empty at the beginning of the game. The step surrounded by a square is included in $G_5$ but not included in $G_4$. 
Game $G_6^A$

$x \leftarrow A^{RO,S^{RO}}$

return $x$

**Procedure $S(0, K, Z)$**

*if* $L_K$ is empty

- $Y_1 || \cdots || Y_\ell \leftarrow RO(K)$ (here, $Y_i \in \{0, 1\}^n$ for each $i$)
- $L_{K, \text{in}} \leftarrow L_{K, \text{in}} \cup \{c_1, \ldots, c_\ell\}$
- $L_{K, \text{out}} \leftarrow L_{K, \text{out}} \cup \{Y_1, \ldots, Y_\ell\}$
- $L_K \leftarrow L_K \cup \{(c_1, Y_1), \ldots, (c_\ell, Y_\ell)\}$

*else if* there exists $Y$ such that $(X, Y) \in L_K$

return $Y$

*else*

- $Y \leftarrow \{0, 1\}^n \setminus L_{K, \text{out}}$
- $L_{K, \text{in}} \leftarrow L_{K, \text{in}} \cup \{X\}$
- $L_{K, \text{out}} \leftarrow L_{K, \text{out}} \cup \{Y\}$
- $L_K \leftarrow L_K \cup \{(X, Y)\}$

**Procedure $S(1, K, Y)$**

*if* $L_K$ is empty

- $Y_1 || \cdots || Y_\ell \leftarrow RO(K)$ (here, $Y_i \in \{0, 1\}^n$ for each $i$)
- $L_{K, \text{in}} \leftarrow L_{K, \text{in}} \cup \{c_1, \ldots, c_\ell\}$
- $L_{K, \text{out}} \leftarrow L_{K, \text{out}} \cup \{Y_1, \ldots, Y_\ell\}$
- $L_K \leftarrow L_K \cup \{(c_1, Y_1), \ldots, (c_\ell, Y_\ell)\}$

*else if* there exists $X$ such that $(X, Y) \in L_K$

return $X$

*else*

- $X \leftarrow \{0, 1\}^n \setminus L_{K, \text{in}}$
- $L_{K, \text{in}} \leftarrow L_{K, \text{in}} \cup \{X\}$
- $L_{K, \text{out}} \leftarrow L_{K, \text{out}} \cup \{Y\}$
- $L_K \leftarrow L_K \cup \{(X, Y)\}$

Figure 7: The ideal game $G_6$ and the simulator $S$. The procedure RO is the same as that of $G_4$ and $G_5$. The procedures $S(0, K, X)$ and $S(1, K, X)$ are described separately so that the notations will be compatible with those in $G_4$ and $G_5$. $S(0, \cdot, \cdot)$ simulates $E(\cdot, \cdot)$ and $S(1, \cdot, \cdot)$ simulates $E^{-1}(\cdot, \cdot)$. 

12
holds since one invocation of $F^E$ makes $\ell$ queries to $E$). Then, for each $i$,

$$\Pr[\text{SetBad}(i)] = \Pr_{Y_1,\ldots,Y_\ell \leftarrow \{0,1\}^n}[Y_j = Y_k \text{ for some } 1 \leq j < k \leq \ell]$$

$$\leq \sum_{1 \leq j < k \leq \ell} \Pr_{Y_j, Y_k \leftarrow \{0,1\}^n}[Y_j = Y_k]$$

$$= \sum_{1 \leq j < k \leq \ell} \sum_{W \in \{0,1\}^\ell \leftarrow \{0,1\}^n} \Pr_{Y_j, Y_k \leftarrow \{0,1\}^n}[Y_j = W \land Y_k = W]$$

$$= \sum_{1 \leq j < k \leq \ell} \sum_{W \in \{0,1\}^\ell \leftarrow \{0,1\}^n} \frac{1}{2^{2n}} \leq \frac{\ell^2}{2^n}$$

holds. Therefore

$$|\Pr[1 \leftarrow G_2^A] - \Pr[1 \leftarrow G_3^A]| \leq \Pr[\text{flag} \leftarrow \text{bad in } G_2]$$

$$\leq \sum_{1 \leq i \leq q_A + \ell Q_A} \Pr[\text{SetBad}(i)]$$

$$\leq \frac{\ell^2 (q_A + \ell Q_A)}{2^n} \quad (3)$$

holds.

From (2) and (3),

$$\text{Adv}_{(F^E, (E, E^{-1})), \text{RO, S}}(A) = |\Pr[1 \leftarrow G_1^A] - \Pr[1 \leftarrow G_0^A]|$$

$$\leq \sum_{1 \leq i \leq 5} |\Pr[1 \leftarrow G_i^A] - \Pr[1 \leftarrow G_{i+1}^A]|$$

$$\leq \frac{\ell^2 (q_A + \ell Q_A)}{2^n}$$

follows.

By definition of the simulator $S$ (Fig. [7]), at each invocation of $S$, it makes at most one query to RO and returns an output in time $O(1)$. Therefore the claim of the theorem holds. $\square$

4 Concluding Remarks

In this paper, we provided a formal security proof of the indifferiability of the SKINNY-HASH internal function (SHI). In the original specification of SKINNY-HASH, the SHI function is claimed to have $n$-bit security without a formal proof. We showed that they are in fact indistinguishable from a random oracle up to $O(2^n)$ queries, as claimed by the designers. Though its construction is quite simple, the SHI function achieves very high security. We hope that more and more function-based sponge constructions will be developed and used in practical situations relying on the SHI function and our security proof.
Acknowledgments

We would like to thank Dr. Donghoon Chang for informing us of the related work [CNL+08]. This work was supported in part by JSPS KAKENHI Grant Number JP20K11675.

References


