Limited-Birthday Distinguishers for Hash Functions

Collisions beyond the Birthday Bound Can Be Meaningful

Mitsugu Iwamoto[1], Thomas Peyrin[2], and Yu Sasaki[3]

[1] Center for Frontier Science and Engineering,
The University of Electro-Communications, Japan
mitsugu@uec.ac.jp

[2] Division of Mathematical Sciences, School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore
thomas.peyrin@gmail.com

[3] NTT Secure Platform Laboratories, Japan
sasaki.yu@lab.ntt.co.jp

Abstract. In this article, we investigate the use of limited-birthday distinguishers to the context of hash functions. We first provide a proper understanding of the limited-birthday problem and demonstrate its soundness by using a new security notion *Differential Target Collision Resistance (dTCR)* that is related to the classical *Target Collision Resistance (TCR)* notion. We then solve an open problem and close the existing security gap by proving that the best known generic attack proposed at FSE 2010 for the limited-birthday problem is indeed the best possible method.

Moreover, we show that almost all known collision attacks are in fact more than just a collision finding algorithm, since the difference mask for the message input is usually fixed. A direct and surprising corollary is that these collision attacks are interesting for cryptanalysis even when their complexity goes beyond the $2^{n/2}$ birthday bound and up to the 2^n preimage bound, and can be used to derive distinguishers using the limited-birthday problem. Interestingly, cryptanalysts can now search for collision attacks beyond the $2^{n/2}$ birthday bound.

Finally, we describe a generic algorithm that turns a semi-free-start collision attack on a compression function (even if its complexity is beyond the birthday bound) into a distinguisher on the whole hash function when its internal state is not too wide. To the best of our knowledge, this is the first result that exploits classical semi-free-start collisions on the compression function to exhibit a weakness on the whole hash function. As an application of our findings, we provide distinguishers on reduced or full version of several hash functions, such as RIPEMD-128, SHA-256, Whirlpool, etc.

Keywords: hash function, compression function, distinguisher, limited-birthday, semi-free-start collision, differential target collision resistance.

K. Sako and P. Sarkar (Eds.) ASIACRYPT 2013, Part II, LNCS 8270, pp. 504–523, 2013.

1 Introduction

A hash function H is a function that takes an arbitrarily long message M as input and outputs a fixed-length hash value of size n bits. Classical security requirements for a cryptographic hash function are collision resistance and (second)-preimage resistance. Namely, it should be impossible for an adversary to find a collision (two distinct messages that lead to the same hash value) in less than $2^{n/2}$ hash computations, or a (second)-preimage (a message hashing to a given challenge) in less than 2^n hash computations. Most standardized hash functions are based upon the Merkle-Damgård paradigm [35,11] and iterate a compression function h with fixed input and output size to handle arbitrarily long messages. The compression function itself should ensure equivalent security properties in order for the hash function to inherit from them. When the internal state size of the compression is the same as for the hash function, then the construction is called *narrow-pipe*, otherwise it is called a *wide-pipe*.

The SHA-3 competition organized by the NIST [49] eventually ended in early October 2012 with the selection of KECCAK [16] as sole winner and new hash function standard. During the last decade, due to this competition and to the cryptanalysis breakthroughs [54,55] that provoked this reaction from the NIST, hash functions have been among the most active topics in academic cryptography. This infatuation is justified by the fact that these primitives are utilized tremendously in practice, with applications ranging from digital signatures, message authentication codes, to secure storage of passwords databases. However, a hash function is also seen as the “swiss knife” of cryptography: many protocols use the random oracle paradigm [3] to check and even prove that they present no structural flaw, and while there is no such thing as a random oracle, designers use hash functions to “simulate” its behavior. Overall, even if collision and (second)-preimage resistance are their most important security properties, cryptographers are therefore also expecting hash functions to present no structural flaw whatsoever, *i.e.* to be indistinguishable from a random oracle. NIST, for example, clearly specified in its SHA-3 call for candidates [49] that the submitted proposals have to support randomized hashing and not present any “non-random behavior”.

On the cryptanalysis side, many various distinguishers have been proposed in the recent years, mainly against AES or SHA-3 candidates. One can cite for example zero-sums distinguishers [2], rotational distinguishers [24] or subspace distinguishers [26]. Limited-birthday distinguishers have been introduced by Gilbert and Peyrin [15] as a tool to distinguish 8 rounds of the AES block cipher from an ideal permutation in the known-key model, and it was later used against other symmetric key primitives [40,37,13,22]. It consists in deriving pairs of plaintext/ciphertext couples $(P, C), (P', C')$ (or input/output couples $(M, H(M)), (M', H(M'))$ for a one-way function) with an input xor difference belonging to a set IN of 2^I elements and an output xor difference belonging to a set OUT of 2^O elements, *i.e.* $P \oplus P' \in IN$ and $C \oplus C' \in OUT$ (or $M \oplus M' \in IN$ and $H(M) \oplus H(M') \in OUT$). What is the best generic attack complexity in the case of an ideal permutation (or function) ? When IN and/or OUT are big enough then this problem is equivalent to a classical birthday paradox problem (*i.e.* with complexity $\min\left\{2^{(n-O)/2}, 2^{(n-I)/2}\right\}$), but the idea underlying the limited-birthday is that when IN and OUT are small an attacker might not be able to use the birthday paradox as much as he would like to. Indeed, he will have to perform several independent smaller birthday searches instead of a single big one, and therefore the process will require much more computations. Gilbert and Peyrin [15] proposed the best known generic algorithm for the limited-birthday problem, whose complexity is $\max\left\{\min\left\{2^{(n-I+1)/2}, 2^{(n-O+1)/2}\right\}, 2^{n-I-O+1}\right\}$ for a permutation and $\max\left\{2^{(n-O+1)/2}, 2^{n-I-O+1}\right\}$ for a function[1]. However, its optimality is yet unknown and it was only conjectured that their attack is the best possible. As of today, only Nikolić *et al.* [39] provided a formal lower bound proof, which is $\min\left\{2^{n/2-2}, 2^{n-(I+O)-3}\right\}$. Unfortunately this bound is not tight and only applies to permutations. For example, in the case of $I = O = 0$, the attack complexity in [15] is $2^{n-I-O+1} = 2^{n+1}$ while the proven bound in [39] only reaches $2^{n/2-2}$.

Some might argue that the limited-birthday problem can trivially be solved by choosing a random input pair (X, Y) and computing $IN = \{X \oplus Y\}$ and $OUT = \{H(X) \oplus H(Y)\}$. However, these pathological attackers, that we call "cheating adversaries", are meaningless: since hash functions are not processing any secret and are completely public (unlike other primitives in cryptography), formalizing security notions requires some kind of challenge, in order to avoid these cheating adversaries (the same is true concerning the chosen-key model for block ciphers). For example, there always exists an adversary that can output a collision with a single operation and negligible memory (i.e. the adversary that just prints a known collision). In general, this obstacle is avoided by considering that a hash function is part of a family indexed by a key input (for example its Initial Value (IV)), or by formalizing the human ignorance [43]. These pathological cases of cheating adversaries are present for all distinguishers without challenges, even for the subspace distinguisher for hash functions [26] or q-multicollisions for block ciphers in the chosen-key model [5].

Our Contributions. To start, we provide in Section 2.1 a proper understanding of limited-birthday distinguishers for the hash function setting. Namely, we discuss potential issues arising from security notions for a public function without challenge and describe various tricks to avoid pathological cheating adversaries. We also show that limited-birthday distinguishers for hash functions can be used to attack a security notion very similar to the classical Target Collision Resistance (TCR) property, which we call differential Target Collision Resistance (dTCR).

Secondly, we provide in Section 2.2 a proof that the currently best known generic attack for the limited-birthday problem (proposed by Gilbert and Peyrin at FSE 2010 [15]) is indeed the best possible. More precisely, we show that the

[1] There is obviously a trade-off between the complexity and the success probability, which here is about 0.63. The original paper [15] missed '+1's in the exponents, which was firstly corrected by [37]

computation complexity to solve the limited-birthday problem is bounded by $\max\left\{2^{(n-O+1)/2}, 2^{n-I-O+1}\right\}$. We can directly conclude that if for a collision attack (*i.e.,* $O = 0$) the set IN of possible message difference of the hash function is limited to one or a few elements regardless of the randomization input, then one can obtain a limited-birthday distinguisher on the function, even with a complexity well beyond the birthday bound. It is to be noted that this condition on the message difference mask is verified for almost all known collision attacks, as for example with the recent advances on SHA-1 [54]. Overall, most known hash function collision attacks are in fact more then just collision finding algorithms since the message difference mask is constrained and, as a consequence, they are now surprisingly becoming interesting even with a complexity beyond the $2^{n/2}$ birthday bound. Our work indicates that concerning distinguishing attacks the security of many hash functions needs to be reevaluated accordingly.

We then move to the case of a compression function, naturally easier to break than the whole hash function. Namely, we provide in Section 3 a generic algorithm that can transform a semi-free-start collision attack on the compression function into a limited-birthday distinguisher for the entire hash function. Because it is based on a meet-in-the-middle approach, this algorithm gets more interesting for the attacker as the internal state of the hash function gets narrower. To the best of the authors knowledge, this conversion is the first result turning a classical semi-free-start collision attack on the compression function into some weakness on the whole hash function (a previous work from Leurent [28] also provides such a conversion, but it is only applicable in the very uncommon case where the average semi-free-start collisions cost is lower than a single operation).

Finally, we provide in Section 4 some applications of our findings against real-world hash functions, such as AES-based hash functions (Section 4.1), HAS-160 (Section 4.2), LANE (Section 4.3), RIPEMD-128 (Section 4.4), SHA-256 (Section 4.5) and Whirlpool (Section 4.6).

2 Limited-Birthday Problem

Throughout this paper, we discuss limited-birthday distinguishers for one-way functions, *i.e.*, in our security model querying input values to obtain the corresponding output values is allowed, but the opposite is forbidden.

In Sect. 2.1, we firstly explain that validating distinguishers without any challenge is hard due to cheating adversaries. We then explain that the ambiguity of the validity does not exist if adversaries are challenged, and the limited-birthday problem is useful even in such a challenged setting. In Sect. 2.2, we formally prove that the previous generic attack that was conjectured as the best attack is indeed optimal. Finally, several remarks are given in Sect. 2.3.

2.1 Importance of the Limited-Birthday Problem in Cryptography

Cheating Adversaries. Collision resistance is the only un-challenged notion of the three classical security properties expected from a cryptographic hash function (collision, preimage, second-preimage), and, as such, the one that proved to be the most difficult to analyze. One of the difficulty that arises for example (and which is true for any un-challenged security property on a public function) is that there is always an adversary that can output a collision immediately, by simply hard-coding it. Rogaway [43] proposed a potential solution to this by formalizing the so-called notion of human ignorance.

However, the existence of another type of pathological cheating adversary has been often utilized as criticism of the limited-birthday distinguishers formalization: the adversary first chooses a random input pair (X, Y), computes $IN = \{X \oplus Y\}$ and $OUT = \{H(X) \oplus H(Y)\}$, and then claims that he can solve the limited-birthday problem with sets IN and OUT (where IN and OUT are actually defined at the end of the attack). It is to be noted that such issues already exist in the case of collision resistance and actually for any security definition regarding a public function with an adversary that is not challenged whatsoever.

Let's come back to our collision resistance case for example. Security engineers obviously understand that collision is an important security definition, but for theoreticians collision is nothing more than a certain output difference Δ which is equal to zero. Collision resistance therefore belongs to a more generic problem that we could name diff(Δ) and which asks for the adversary to exhibit an input pair (X, Y) such that $H(X) \oplus H(Y) = \Delta$. All members of this set are equally hard with regards to generic attacks. Collision resistance is actually diff(0), but cheating adversaries exist for diff(Δ): by just choosing a random input pair (X, Y) and trivially claiming that we can solve diff($H(X) \oplus H(Y)$).

Similarly, one can design cheating adversaries for the recent q-multicollision problem [5] used on AES: define the problem q-multi-diff($\Delta_1, \ldots, \Delta_q$) that asks for the attacker to exhibit q input pairs $(X_1, Y_1), \ldots, (X_q, Y_q)$ such that $H(X_1) \oplus H(Y_1) = \delta \oplus \Delta_1, \ldots, H(X_q) \oplus H(Y_q) = \delta \oplus \Delta_q$. Then the q-multicollision problem is nothing else than q-multi-diff($0, \ldots, 0$) with a predefined δ, yet obvious cheating adversaries exist for q-multi-diff: just pick q random input pairs $(X_1, Y_1), \ldots, (X_q, Y_q)$, and claim that you can solve q-multi-diff($H(X_1) \oplus H(Y_1) \oplus \delta, \ldots, H(X_q) \oplus H(Y_q) \oplus \delta$). The same reasoning applies to the subspace distinguishers [26] as well.

As a direct analogy, the limited-birthday problem LBP(IN, OUT) with fully defined sets IN and OUT belongs to the more general limited-birthday problem LBP. Thus, the limited-birthday distinguishers are as valid as collision, q-multicollision or subspace distinguishers when the sets IN and OUT are fully defined, and we emphasize that in the rest of the article the sets IN and OUT are considered to be fully defined before the attacker starts to actually search for a valid pair of inputs. Yet, in addition, we propose below some solutions to overcome any potential cheating adversaries.

Challenging the Adversary. There are several cryptographic protocols that allow users to provide some tweak to a function H. The tweak, T, plays the role of enhancing the security, *i.e.*, the attacker cannot obtain the target function

H_T until the tweak value is determined. The limited-birthday distinguisher is particularly useful for evaluating such a tweakable function H_T. One of such protocols is the randomized hashing [50], where a message to be signed with a digital signature scheme is hashed after a tweak is applied in order to enhance the security against forgery attacks. Let us first recall the security notion called *target collision resistance* [4]. An n-bit tweakable function H_T is said to be target collision resistant if it is computationally hard to perform the following attack.

Target Collision Resistance (TCR)

1. *The adversary chooses an input value I after some precomputation.*
2. *The value of T is chosen without any control by the adversary.*
3. *The adversary finds an input value $I \oplus \Delta$ such that $H_T(I) = H_T(I \oplus \Delta)$.*

The *TCR* notion is a base of the provable security of the randomized hashing scheme[2]. In the SHA-3 competition, NIST required the submitted algorithms to provide n bits of security for the randomized hashing scheme [49, Section 4.A]. We then slightly modify the *TCR* notion as follows.[3]

Differential Target Collision Resistance (dTCR)

1. *The adversary chooses an input difference Δ after some precomputation.*
2. *The value of T is chosen without any control by the adversary.*
3. *The adversary finds an input value I such that $H_T(I) = H_T(I \oplus \Delta)$.*

Let the tweak T be a choice of a part of the algorithm design such as constant values, Sboxes, and IV. For such a tweak, a differential attack can usually choose IN and OUT independently of T. Therefore, for such a tweak, a limited-birthday distinguisher for the hash function setting with $|IN| = 1, OUT = \{0\}$, and with a complexity below 2^n, is an attack on the *dTCR* notion. In section 4, we will show several applications to real-world hash functions that satisfy those properties against the tweaking method of the randomized hashing. We believe that the impact of limited-birthday distinguishers is much bigger than just identifying a non-random behavior as several other distinguishers do.

In the case of iterative hash functions, a very simple tweak can even be considered: randomizing the first message block M_1. The attacker is challenged to exhibit a non-random property on the function and with M_1 as prefix chosen by the challenger, *i.e.* every message queried or used must contain message block M_1 as prefix. In fact, the randomized hashing gives a tweak by choosing a random string r, and processing r as a prefix and then XORing r to each input message block. Because a challenge is asked to the attacker preliminarily, no cheating

[2] Strictly speaking, security of the randomized hashing scheme is based on the *eTCR* notion [18], for which the adversary finds input values $(T', I \oplus \Delta)$ such that $F_T(I) = F_{T'}(I \oplus \Delta)$ at Step 3 of the definition of *TCR*. Note that breaking *TCR* immediately leads to breaking *eTCR*.

[3] The two notions are similar, yet we leave as open problem the question regarding any formal link between them.

adversary exists in this setting. Moreover, many differential attacks can find IN and OUT independently of the tweak value.

Note also that it is important for the tweak set size to be big enough, in order to avoid any adversary that would precompute cheating behavior for any tweak value.

2.2 The Limited-Birthday Problem for Hash Functions

Definition 1 (The limited-birthday problem). *Let H be an n-bit output hash function, that can be randomized by some input (IV or tweak or etc.) and that processes input messages of fixed size, m bits where $m \geq n$. Let IN be a set of admissible input differences and OUT be a set of admissible output differences, with the property that IN and OUT are* ***closed sets*** *with respect to $\oplus$. Then, for the limited-birthday problem, the goal of the adversary is to generate a message pair (M, M') such that $M \oplus M' \in IN$ and $H(M) \oplus H(M') \in OUT$ for a randomly chosen instance of H.*

A generic procedure to solve the limited-birthday problem in [15] is described below. We denote by *active* (resp. *inactive*) the input bits for which the xor difference *cannot* be chosen by the attacker (resp. *can* be chosen by the attacker). Its illustration is given in Figure 2 in Appendix.

1. Choose a random value for the inactive bits.
2. For all $|IN|$ values of the active bits, call the function oracle and obtain the corresponding output values. Then, build $\binom{|IN|}{2} \approx |IN|^2/2$ pairs with the queries replies received.
3. If a pair whose output difference is included in OUT is found, abort the procedure. Otherwise, go back to Step 1 and choose another random value for the inactive bits.

Note that if $\binom{|IN|}{2} > 2^n/|OUT|$, choosing $\sqrt{2^{n+1}/|OUT|}$ values of active bits in Step 2 is enough.

Theorem 1. *The limited-birthday attack complexity in [15] for a one-way function is*

$$\max\left\{\sqrt{\frac{2^{n+1}}{|OUT|}}, \frac{2^{n+1}}{|IN| \cdot |OUT|}\right\} = \max\left\{2^{\frac{n-O+1}{2}}, 2^{n-I-O+1}\right\} \qquad (1)$$

where I and O are defined by $|IN| = 2^I$ and $|OUT| = 2^O$, respectively.

If $|IN|$ is small, the complexity is $2^{n-I-O+1}$. However, even if $|IN|$ is very big, the complexity cannot be below $2^{\frac{n-O+1}{2}}$. Thus, the complexity is the maximum of these two cases. It was conjectured that the above attack procedure is the best possible. Then, based on this conjecture, presenting for a real hash function an attack which is faster than Eq. (1) was regarded as a non-ideal behavior and many results have been published in this context [15,40,13,22]. We close an open problem by proving below the optimality of the above generic limited-birthday attack.

Theorem 2. *The lower bound of the number of queries for the limited-birthday distinguisher matches Eq. (1).*

Proof. Let U be the attack complexity, *i.e.* the number of queries for the limited-birthday distinguisher. In the case of $\binom{2^{n-I}}{2} > 2^{n-O}$, it holds that $U \geq 2^{\frac{n-O+1}{2}}$ since, in this case, the situation is equivalent to the ordinary birthday attack. Hence, it is sufficient to prove that $U \geq 2^{n-I-O+1}$ in the case of $\binom{2^{n-I}}{2} \leq 2^{n-O}$.

First, let $\mathcal{I} := \{1, 2, \ldots, 2^{n-I}\}$ and $\mathcal{O} := \{1, 2, \ldots, 2^{n-O}\}$ represent the sets of inactive bits in inputs and outputs, respectively, and fix a set of queries by the limited-birthday distinguisher *arbitrarily.* According to this set of queries, a bipartite graph $G := (\mathcal{I}, \mathcal{O}, E)$ can be defined as shown in Figure 1, where $\mathcal{I}$ and $\mathcal{O}$ are partite sets and E is the edge set. In the bipartite graph G, each edge $e := (i, j) \in E$, $i \in \mathcal{I}$, $j \in \mathcal{O}$, corresponds to a query with an inactive bit $i \in \mathcal{I}$ and its output $j \in \mathcal{O}$. Due to this correspondence, the bipartite graph G allows multiedges which share the same end vertices. The pair of queries satisfying limited-birthday collision corresponds to the multiedges, which we are going to find.

Hereafter, we call a pair of edges which share the same vertex in $\mathcal{I}$ (but *no* constraint for the other end vertex in $\mathcal{O}$) as *a valid pair.* Because, for each edge, the end vertex belonging to $\mathcal{O}$ is chosen according to the uniform distribution, the probability that a randomly chosen valid pair is a solution for the limited-birthday problem is $2^{-(n-O)}$. Therefore, the total number of valid pairs, denoted by V, should be greater than or equal to 2^{n-O} in order to obtain a solution for the limited-birthday problem with a good probability.

For $i \in \mathcal{I}$, let d_i be the degree of the vertex i, which is the number of edges connected to the vertex i. It is obvious that d_i is no more than 2^I, and the number of valid pairs incident with the vertex i is $\binom{d_i}{2}$. Hence, the total number V of valid pairs can be expressed as

$$V = \sum_{i=1}^{2^{n-I}} \binom{d_i}{2} \approx \frac{1}{2} \sum_{i=1}^{2^{n-I}} d_i^2. \tag{2}$$

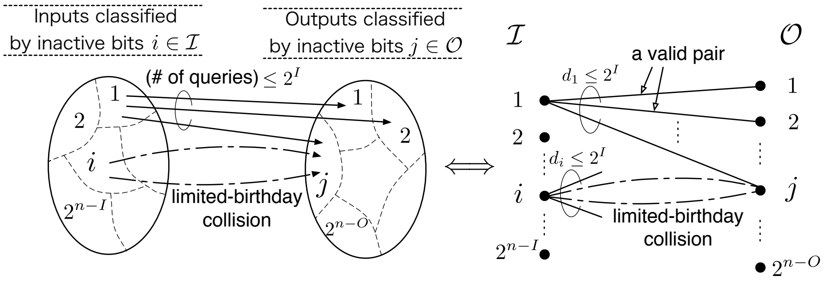

Inputs classified by inactive bits $i \in \mathcal{I}$
Outputs classified by inactive bits $j \in \mathcal{O}$
(# of queries) $\leq 2^I$
1
2
i
2^{n-I}
1
2
j
2^{n-O}
limited-birthday collision
$\Longleftrightarrow$
$\mathcal{I}$
$\mathcal{O}$
a valid pair
$d_1 \leq 2^I$
$d_i \leq 2^I$
1
2
i
2^{n-I}
1
2
j
2^{n-O}
limited-birthday collision

Fig. 1. Graph representation of general strategy of limited-birthday attacks

Noticing that the degree of each vertex belonging to $\mathcal{I}$ can have at most 2^I and the total number of queries is U, we have the following constraints without loss of generality:

$$\sum_{i=1}^{2^{n-I}} d_i = U; \qquad 2^I \geq d_1 \geq d_2 \geq \cdots \geq d_{2^{n-I}} \geq 0. \tag{3}$$

Here, we also note that the above $(d_1, d_2, \ldots, d_{2^{n-I}})$ is determined by the set of queries by the distinguisher, namely, it can represent *arbitrary* attack strategy including the limited-birthday attack proposed in [15]. Hence, the best possible attack can be obtained by maximizing the total number of valid pairs V.

In order to maximize V in Eq. (2) under the constraints Eq. (3), theory of majorization is useful [30]: for real valued ℓ-dimensional vectors $\boldsymbol{x} = (x_1, x_2, \ldots, x_\ell) \in \mathbb{R}^\ell$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_\ell) \in \mathbb{R}^\ell$ arranged as decreasing order, *i.e.* $x_1 \geq x_2 \geq \cdots \geq x_\ell$ and $y_1 \geq y_2 \geq \cdots \geq y_\ell$, we say that $\boldsymbol{y}$ is majorized by $\boldsymbol{x}$, in symbols $\boldsymbol{x} \succ \boldsymbol{y}$, if they satisfy $\sum_{i=1}^t x_i \geq \sum_{i=1}^t y_i$ for $1 \leq t \leq \ell - 1$ and $\sum_{i=1}^\ell x_i = \sum_{i=1}^\ell y_i$. We note that a function $f : \mathbb{R}^\ell \to \mathbb{R}$ is said to be Schur-convex if $f(\boldsymbol{x}) \geq f(\boldsymbol{y})$ is satisfied for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^\ell$ with $\boldsymbol{x} \succ \boldsymbol{y}$. It is well known[4] that a function $\sum_{i=1}^\ell x_i^k$ is Schur-convex on $\mathbb{R}_+^\ell$ for any $k > 1$.

Based on theory of majorization, the vector $\boldsymbol{D^*} = (d_1^*, d_2^*, \ldots, d_{2^{n-I}}^*)$ defined by[5]

$$d_i^* = \begin{cases} 2^I, & \text{for} \quad 1 \leq i \leq U/2^I \\ 0, & \text{for} \quad U/2^I < i \leq 2^{n-I} \end{cases} \tag{4}$$

attains the maximum value of V under the constraints of Eq. (3). To see this, it is sufficient to check that the vector $\boldsymbol{D}^*$ majorizes all vectors satisfying Eq. (3), and the fact that the function $\sum_{i=1}^n x_i^2$ is Schur-convex. Hence, substituting Eq. (4) into (3), we can upper-bound V as

$$V \leq \frac{1}{2} \cdot 2^{2I} \cdot \frac{U}{2^I} = \frac{U \cdot 2^I}{2}. \tag{5}$$

As we have already seen, $V \geq 2^{n-O}$ is necessary in order to find a limited-birthday collision with sufficiently high probability. Combining this inequality with Eq. (5), we obtain $U \geq 2^{n-I-O+1}$, which completes the proof. □

2.3 Remarks

The proof in Section 2.2 can be extended to the lower bound of the query complexity for the 4-sum, or in general the k-sum problem, with pre-specified admissible difference sets IN. Here, the k-sum problem finds k distinct input values where the xor sum of their output values is 0. It is already known that

[4] For instance, this fact is immediately recognized from [30, C.1. Proposition] which states that $\sum_i g(x_i)$ is Schur convex if $g(x)$ is convex. Obviously, $g(x) = x^k$, $x \geq 0$, is convex for any $k > 1$.

[5] We roughly assume that U is a power of 2.

several signature schemes [52] and several instantiations of the random oracle [29] are badly affected if an underlying hash function is vulnerable against the k-sum attack. When the degree of each input vertex is d_i in Figure 1, the number of valid k-tuples of edges that share the same input vertex is $\sum_{i=1}^{2^{n-I}} \binom{d_i}{k}$, which is approximately $(1/k!) \cdot \sum_{i=1}^{2^{n-I}} d_i^k$. Because the function $\sum_i d_i^k$ for any $k > 1$ is Schur-convex, We can prove that $\boldsymbol{D}^*$ which majorizes any other 2^{n-I}-dimensional vectors is the optimal choice to minimize the query complexity.

Finally, it is to be noted that the reasoning of our proof is only done on the input and output set sizes. Therefore, one can use this proof even for other properties than xor difference. When IN and/or OUT are not closed sets our proof still applies, but is not tight since the algorithm from [15] can not be utilized anymore. We leave this gap as an open problem, yet conjecturing that the attack complexity will grow rapidly as the sets gets more opened.

3 Generic Limited-Birthday Distinguishers

Several previous works analyzed the complex relation between the security of a hash function and its compression function, both in a proof oriented [9] or in an attack oriented manner [41]. For example, a well known result is that a preimage attack for a compression function (also called pseudo-preimage attack) can be transformed into a preimage attack on the hash function when a narrow-pipe design is used by a meet-in-the-middle technique. In this section, we explain how an attacker can turn a semi-free-start collision attack (even when its complexity is beyond the birthday bound) into a limited-birthday distinguisher on the hash function using a meet-in-the-middle approach.

Let h be a compression function taking m bits of message and k bits of chaining variable as inputs and outputting a k-bit value. Then, let H be an n-bit hash function (with $n \leq k$), that iteratively calls h to process incoming m-bit message words. A semi-free-start collision is a pair $((CV, M), (CV, M'))$ with $M \neq M'$ and such that $h(CV, M) = h(CV, M')$. We assume that an attacker is able to find 2^s distinct semi-free-start collisions for h with complexity 2^c operations (by distinct we mean that at least each CV value is different), with $s \leq k/2$. Let IN be the set of the possible message difference masks for all these semi-free-start collisions, and we still denote its size by $|IN| = 2^I$. We derive a limited-birthday distinguisher on H with a simple meet-in-the-middle technique as follows:

1. generate the 2^s semi-free-start collisions $((CV_j, M_j), (CV_j, M'_j))$ on h with 2^c operations and add all 2^s CV_j values in a list L
2. from the hash function initial value IV, pick 2^{k-s} random message blocks M_i. Compute their corresponding output value after application of h and place these values in a list L'.
3. check if there is a collision between a member of L and L', and output as solution the corresponding input message couple $((M_i||M_j), (M_i||M'_j))$, that verifies $H(M_i||M_j) = H(M_i||M'_j)$. Note that collisions are propagated when adding extra message blocks in the hash computation chain, thus the padding constraint is always satisfied.

First, it is clear that during the third phase we have enough elements in both lists (2^{k-s} and 2^s) to find a collision with good probability. The overall complexity is $2^c + 2^{k-s}$ operations and $\min\{2^{k-s}, 2^s\}$ memory.[6] The attacker outputs a collision for the hash function (fixed output difference mask to zero, thus $|OUT| = 1$) with an input difference mask lying in a space IN of size 2^I (since the IV of the hash function is fixed for both members of the pair and since the difference mask zero is applied to the first block M_i), and the limited-birthday tells us that this should cost $\max\{2^{n/2}, 2^{n-I+1}\}$ in the ideal case. Since $2^c + 2^{k-s} \geq 2^s + 2^{k-s} \geq 2^{k/2} \geq 2^{n/2}$, this attack will lead to a valid distinguisher if and only if

$$2^c + 2^{k-s} < 2^{n-I+1}. \tag{6}$$

One may wonder why we do not simply use a parameter $x = c - s$ that represents the average semi-free-start collision cost instead of c and s (and then the attack complexity would simply be $2^{(k+x)/2+1}$). The reason is that many semi-free-start collision attacks consume a lot of freedom degrees and often the attacker is unable to generate as many as he wants. Looking at the relation (6), one can remark that for a particular hash function (*i.e.* k and n are fixed) and for a fixed I, the attacker only has to find the right amount of semi-free-start collisions that minimizes $2^c + 2^{k-s}$. Also, in the best case where a semi-free-start collision costs a single operation on average (i.e. $c = s$), the best for him is to generate as many semi-free-start collisions as he can (up to $2^{k/2}$). More generally, the cheaper are the semi-free-start collisions to generate, the closer the distinguisher will be to the $2^{k/2}$ birthday bound. Conversely, the more expensive are semi-free-start collisions to generate, the closer the distinguisher will be to the 2^k internal preimage bound. Finally, because of its meet-in-the-middle nature, it is only natural that the complexity of the attack reduces when the size of the hash function internal pipe decreases. For hash candidates with double-pipe and more ($k \geq 2n$), our algorithm will never lead to a valid distinguisher, which is yet another argument indicating that having at least a double-pipe for a hash function increases its security.

It is to be noted that the very same reasoning can be applied even if the semi-free-start collision attack requires several message blocks in order to be performed. Moreover, one can even further generalize by looking at semi-free-start near-collision attacks, that is finding a pair $((CV, M), (CV, M'))$ with $M \neq M'$ and such that $h(CV, M) \simeq h(CV, M')$. However, near collisions (unlike real collisions) do not propagate when adding extra message blocks in the hash computation chain. Therefore, in order to use semi-free-start near-collision attacks, it is necessary that they have to be able to include the hash padding inside the

[6] If the cost for generating each semi-free-start collision is 1, the matching process becomes the balanced meet-in-the-middle, and thus a memoryless attack might be possible with a cycle method. However, in order to construct the cycle, one must define how to make the feed for the next computation and the feasibility will depend on the details of the semi-free-start collision attack.

last message block. Then, the only effect compared with previous reasoning will be that $|OUT|$ will be slightly larger than 1.

This method shows that semi-free-start collisions on a compression function are directly meaningful even for the hash function security itself. Even better, cryptanalyst might now be interested in finding semi-free-start collision attacks beyond the birthday bound, in order to derive distinguishers on the entire hash function. Previously, Leurent [28] also used a meet-in-the-middle technique on Skein [14] to turn semi-free-start collisions into a collision on the whole hash, but his method is only applicable in the uncommon situation where the average cost of the semi-free-start collisions is strictly lower than 1 (in his article 2^{70} semi-free-start collisions can be generated with 2^{40} operations).

Finally, one may argue that distinguishers from a random oracle already existed for classical iterative hash functions with a rather narrow-pipe, for example by using the very simple and well known length extension attack (for all Z, from $H(M_1||\dots||M_i)$ one can compute the value of $H(M_1||\dots||M_i||Z)$, without even knowing $M_1||\dots||M_i$). However, such issues do not exist anymore for strengthen constructions like the ones proposed by Coron *et al.* [9]. For example, utilizing a HMAC-like construction (like it is done in the LANE hash function [20]) prevents the length extension attack, while our limited-birthday distinguishing attack would remain perfectly valid.

4 Applications

In this section, we show a few application examples of our generic hash function limited-birthday distinguisher from compression function semi-free-start collisions. While some of the results we will present here are quite interesting such as the first result on the full LANE hash function and improved results on RIPEMD-128 and Whirlpool, some other do not reach the full number of rounds or do not really improve over known distinguishers. However, we emphasize that due to the tremendous work required to analyze the collision resistance of a compression function, we mostly based our application examples on known semi-free-start collision attacks. Therefore, since beyond-birthday complexity semi-free-start collisions were not searched for so far, we expect that several of our results can be improved by allowing this extra complexity cost. We summarize our distinguishers in Table 1. The limited-birthday distinguisher on the hash function with $|IN| = 1$, $OUT = \{0\}$ can be used to attack the *dTCR* notion against the randomized hashing. Our results on HAS-160, RIPEMD-128, and SHA-256 are the cases.

4.1 Reduced-Round AES-Based Hash Functions

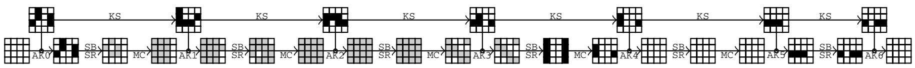
AES-128 [10] is a 128-bit block cipher with 128-bit keys and the NIST's current block cipher standard. It is composed of 10 rounds (in the last round, the linear diffusion layer is removed) and many recent hash functions got inspired by this design. Classic ways to securely turn a block cipher E into a

Table 1. Summary of the new results for the limited-birthday distinguishers on various hash functions

target	rounds	time	memory	type	source
AES-DM hash func.	7/10	2^{125}	2^{8}	preimage attack	[44]
AES-DM hash func.	6/10	2^{113}	2^{32}	limited-birthday dist.	Sect. 4.1
AES-MP hash func.	7/10	2^{120}	2^{8}	2nd preimage attack	[44]
AES-MP hash func.	6/10	2^{89}	2^{32}	limited-birthday dist.	Sect. 4.1
HAS-160 hash func.	68/80	$2^{156.3}$	2^{15}	preimage attack	[19]
HAS-160 hash func.	65/80	2^{81}	2^{80}	limited-birthday dist.	Sect. 4.2
LANE-256 hash func.	full	2^{169}	2^{88}	limited-birthday dist.	Sect. 4.3
LANE-512 hash func.	full	2^{369}	2^{144}	limited-birthday dist.	Sect. 4.3
RIPEMD-128 hash func.	full	$2^{105.4}$	negl.	limited-birthday dist.	[27]
RIPEMD-128 hash func.	full	$2^{95.8}$	$2^{33.2}$	limited-birthday dist.	Sect. 4.4
SHA-256 hash func.	42/64	$2^{251.7}$	negl.	preimage attack	[1]
SHA-256 hash func.	38/64	2^{129}	2^{128}	limited-birthday dist.	Sect. 4.5
Whirlpool hash func.	6/10	2^{481}	2^{256}	preimage attack	[26]
Whirlpool hash func.	7/10	2^{440}	2^{128}	limited-birthday dist.	Sect. 4.6

compression function h are known for a long time *e.g.,* the Davies-Meyer mode ($h(CV, M) = E_M(CV) \oplus CV$) or the Miyaguchi-Preneel mode ($h(CV, M) = E_{CV}(M) \oplus M \oplus CV$). Concretely, we will consider compression functions built upon AES-128 in these two modes, and placed into a Merkle-Damgård domain extension to obtain the hash function. This was actually a proposal by Cohen [8] and the current best attack on the whole hash function is a 7-round preimage attack [44], but with a complexity very close to the generic one. In this Section, we will consider truncated differential paths and denote an active/inactive byte by a black/white cell.

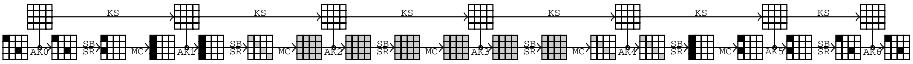
Davies-Meyer Mode: we use the following 6-round truncated differential path:



The differential path in the key schedule can be handled independently from the internal cipher part, and the cost is very low (only 6 Sbox transitions to control). Using the Super-Sbox technique from [15,26], one can derive a pair verifying the 3 middle-left rounds part (light gray cells) with complexity 1 on average. The rest of the truncated differential path is verified probabilistically forward and backward from this middle part. 5 Sbox differential transitions have to be controlled on the left, $8 + 3 = 11$ have to be controlled on the right, and for each transition we can use the best 2^{-6} transition probability of the AES Sbox. Therefore, the uncontrolled part of the differential path will be verified with probability 2^{-96} and one solution for the entire path (i.e. a semi-free-start in the Davies-Meyer mode) can be found with complexity 2^{96}.

Using parameters $n = k = 128$, $c = 112$ and $s = 16$ for our conversion algorithm, we obtain a hash function limited-birthday distinguisher complexity of 2^{113} computations. Since difference on the input message of the compression function is fully defined, we have $I = 0$ and our limited-birthday proof tells us that the complexity for an ideal function is 2^{129}. A basic freedom degrees evaluation shows that one can generate much more semi-free-start collisions that required.

Miyaguchi-Preneel Mode: we use the following 6-round truncated differential path:



Using the Super-Sbox technique, one can derive a pair verifying the 3 middle rounds part (light gray cells) with complexity 1 on average. The rest of the path is verified probabilistically, with probability 2^{-32} (two MixColumns transitions from 4 to 2 active bytes). Therefore, one solution for the entire path can be found with complexity 2^{32} and obtaining a collision at the output of the Miyaguchi-Preneel mode requires an extra 2^{16} for a total complexity of 2^{48} computations.

Using parameters $n = k = 128$, $c = 88$ and $s = 40$, we obtain a hash function limited-birthday distinguisher complexity of 2^{89} computations. Since the input message can contain only one byte of random difference we have $I = 8$ and our limited-birthday proof tells us that the complexity for an ideal function is $2^{128-16+1} = 2^{113}$. Note that freedom degrees not a problem since we choose any key value and for each key we expect about 2^8 semi-free-start collisions.

4.2 Reduced-Round HAS-160

HAS-160 is a hash function standardized by the Korean government and widely used in Korea [48]. Its structure is similar to SHA-1. It adopts the narrow-pipe Merkle-Damgård structure, and produces 160 bits digests. The compression function consists of 80 steps.

Although a distinguisher on the full compression function is known [45], the current best attack for the hash function is a 68-step preimage attack proposed by Hong *et al.* [19], which is slightly faster than the brute force attack. For a practical complexity, a semi-free-start collision attack for 65 steps of the compression function was proposed by Mendel *et al.* [33].

The attack in [33] can generate a semi-free-start collision with complexity 1. Moreover, the attack has enough amount of freedom degrees to generate many semi-free-start collisions. Using parameters $n = 160, k = 160, c = 80$ and $s = 80$, the distinguisher on the hash function can be mounted with a complexity of 2^{81} compression function computations and 2^{80} memory. Since the differential mask on the message input is fully fixed, we have $I = 0$ and the generic complexity to solve this limited-birthday instance is 2^{161} computations, which validates our distinguisher.

4.3 LANE

LANE was designed by Indesteege [20] and submitted to the NIST's SHA-3 competition. Although LANE did not make it to the second round of the process, no security weakness has been discovered yet on the hash function. It adopts a narrow-pipe Merkle-Damgård like structure.

The current most significant attack on LANE is a semi-free-start collision attack on the full compression function by Matusiewicz *et al.* [32] and its improvement by Naya-Plasencia [36], which generates semi-free-start collisions for LANE-256 and LANE-512 with 2^{80} and 2^{224} compression function computations respectively and a memory to store 2^{66} states.

By using our conversion method, this semi-free-start collision attack on the compression function can be converted into a distinguisher on the entire hash function (which tends to indicates thus it was eventually a wise move from NIST to remove this candidate from the competition). Having no strong restriction on the amount of freedom degrees, with parameters $n = k = 256$, $c = 168$ and $s = 88$, the complexity of our distinguisher for LANE-256 is 2^{169} compression function computations and 2^{88} memory. On the other hand, the semi-free-start collision attack accepts any difference on 10 fixed byte positions, which gives us $I = 80$. Our limited-birthday proof tells us that the complexity for an ideal function is $2^{256-80+1} = 2^{177}$, which validates our attack.

Regarding LANE-512, by choosing parameters $n = k = 512$, $c = 368$ and $s = 144$, we minimize the distinguisher complexity to 2^{369} computations and 2^{144} memory. On the other hand, the semi-free-start collision attack accepts any difference on 16 fixed byte positions, which gives us $I = 128$. Our limited-birthday theorem tells us that the complexity for an ideal function to find this input pair is $2^{512-128+1} = 2^{385}$, which validates our attack.

4.4 RIPEMD-128

RIPEMD-128 [12] is a 128-bit hash function (standardized at ISO/IEC [21]) that uses the Merkle-Damgård construction and whose compression function has the particularity to use two parallel computation branches. Semi-free-start collisions on the compression function can be generated with $2^{61.6}$ computations and negligible memory as shown recently [27]. Moreover, a distinguisher on the full hash function was also proposed in the same article, requiring $2^{105.4}$ computations.

Using our conversion algorithm, we utilize the semi-free-start collision attack to derive a limited-birthday distinguisher. Namely, using parameters $n = k = 128$, $c = 94.8$ and $s = 33.2$, we obtain a distinguisher complexity of $2^{95.8}$ computations and $2^{33.2}$ memory (about $2^{33.2}$ semi-free-start collisions need to be generated, which seems to not be an issue as the authors of [27] analyzed that a lot of freedom degrees were available). Since the differential mask on the message input for the semi-free-start collision attack is fully fixed, we have $I = 0$ and the generic complexity to solve this limited-birthday instance is 2^{129} computations, which validates our distinguisher.

4.5 Reduced-Round SHA-256

SHA-256 [51] is one of the NIST approved hash functions. It is a narrow-pipe 256-bit hash function that uses the Merkle-Damgård construction and whose compression function is composed of 64 rounds. Recently, a semi-free-start collision attack on 38-round reduced SHA-256 compression function has been proposed [34] with a complexity equivalent to 2^{37} computations. However, once a semi-free-start collision has been found many can be obtained for free, providing an average cost of a single operation per solution. The currently best known attack on the hash function is a preimage attack [1] on 42 rounds with complexity $2^{251.7}$ computations.

We utilize the semi-free-start collision attack to derive a limited-birthday distinguisher. Namely, using parameters $n = k = 256$, $c = 128$ and $s = 128$, we obtain a distinguisher complexity of 2^{129} computations and 2^{128} memory (about 2^{128} semi-free-start collisions need to be generated in our case, which is possible when studying the differential path provided in [34]). Since the differential mask on the message input for the semi-free-start collision attack is fully fixed, we have $I = 0$ and the generic complexity to solve this limited-birthday instance is 2^{257} computations, which validates our distinguisher.

4.6 Reduced-Round Whirlpool

Whirlpool [42] is a 512-bit hash function proposed by Rijmen and Barreto in 2000. which was standardized by ISO [21] and recommended by NESSIE [38]. The compression function consists of a 10-round AES-based cipher in a Miyaguchi-Preneel mode and whose key schedule also consists of AES-like rounds. The current best attack in the hash function setting is a 6-round preimage attack by Sasaki *et al.* [46]. Lamberger *et al.* presented a 7-round near-collision attack [26]. Although it can handle the fixed IV, the attack cannot satisfy the padding constraint and thus does not apply on the full hash function.

We propose a 7-round distinguisher by using our conversion method. The base of our distinguisher is a semi-free-start collision attack for 7 rounds of the Whirlpool compression function proposed by Lamberger *et al.* [26], which requires 2^{128} compression function computations and memory to store 2^{128} states to generate a semi-free-start collision. However, the amount of freedom degrees only allows to generate 2^{72} solutions and once a precomputation table with 2^{128}

entries is built, the average complexity of generating a semi-free-start collision is 2^{120}, not 2^{128}. Therefore, we have parameters $n = k = 512$, $c = 192$ and $s = 72$ for our limited-birthday distinguisher.

The attack complexity is then 2^{440} computations and 2^{128} memory. Since for Lamberger *et al.*'s attack, only a single byte will contain an uncontrolled difference, we have $I = 8$ and the limited-birthday proof tells us that in the ideal case finding such a pair should cost 2^{505} computations.

5 Conclusion

In this article, we have explored the limited-birthday distinguishers for the case of hash functions. We believe that this type of distinguishers is powerful, and will provide new insights on how hash functions can simulate random oracles in practice. Surprisingly, on both the hash or the compression function, cryptanalysts can now look for collision attacks beyond the birthday bound and up to the preimage bound. Finally, our conversion algorithm is yet another argument in favor of long-pipe hash functions, which seems to be a good protection against compression function weaknesses turning into hash function weaknesses.

As future work, we leave the security proofs for the permutation case as an open problem. It would also be worth analyzing other types of distinguishers, such as the ones based on integral attacks [25], and try to derive better lower bounds for the ideal case. Obviously, on the cryptanalysis side, it would interesting to see how far can the limited-birthday distinguishers go for high-end hash functions, and in particular to what extent can the known (semi)-free-start collision attacks be extended, by allowing the attacker a computation limit up to the preimage bound.

Acknowledgments. The authors would like to thank the anonymous referees for their helpful comments. Mitsugu Iwamoto is supported by JSPS KAKENHI Grant Number 23760330. Thomas Peyrin is supported by the Singapore National Research Foundation Fellowship 2012 (NRF-NRFF2012-06).

References

1. Aoki, K., Guo, J., Matusiewicz, K., Sasaki, Y., Wang, L.: Preimages for Step-Reduced SHA-2. In: Matsui (ed.) [31], pp. 578–597
2. Aumasson, J.-P., Meier, W.: Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi (2009)
3. Bellare, M., Rogaway, P.: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In: ACM Conference on Computer and Communications Security, pp. 62–73 (1993)
4. Bellare, M., Rogaway, P.: Collision-Resistant Hashing: Towards Making UOWHFs Practical. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 470–484. Springer, Heidelberg (1997)
5. Biryukov, A., Khovratovich, D., Nikolic, I.: Distinguisher and Related-Key Attack on the Full AES-256. In: Halevi (ed.) [17], pp. 231–249

6. Brassard, G. (ed.): CRYPTO 1989. LNCS, vol. 435. Springer, Heidelberg (1990)
7. Canteaut, A. (ed.): FSE 2012. LNCS, vol. 7549. Springer, Heidelberg (2012)
8. Cohen, B., Laurie, B.: AES-hash. Submission to NIST: Proposed Modes (2001), http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/aes-hash/aeshash.pdf
9. Coron, J.-S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård Revisited: How to Construct a Hash Function. In: Shoup (ed.) [47], pp. 430–448
10. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer (2002)
11. Damgård, I.: A Design Principle for Hash Functions. In: Brassard (ed.) [6], pp. 416–427
12. Dobbertin, H., Bosselaers, A., Preneel, B.: RIPEMD-160: A Strengthened Version of RIPEMD. In: Gollmann, D. (ed.) FSE 1996. LNCS, vol. 1039, pp. 71–82. Springer, Heidelberg (1996)
13. Duc, A., Guo, J., Peyrin, T., Wei, L.: Unaligned Rebound Attack: Application to Keccak. In: Canteaut (ed.) [7], pp. 402–421
14. Ferguson, N., Lucks, S., Schneier, B., Whiting, D., Bellare, M., Kohno, T., Callas, J., Walker, J.: The Skein Hash Function Family. Submission to NIST (Round 3) (2010)
15. Gilbert, H., Peyrin, T.: Super-Sbox Cryptanalysis: Improved Attacks for AES-Like Permutations. In: Hong, S., Iwata, T. (eds.) FSE 2010. LNCS, vol. 6147, pp. 365–383. Springer, Heidelberg (2010)
16. Peeters, M., Bertoni, G., Daemen, J., Van Assche, G.: The Keccak SHA-3 submission. Submission to NIST, Round 3 (2011)
17. Halevi, S. (ed.): CRYPTO 2009. LNCS, vol. 5677. Springer, Heidelberg (2009)
18. Halevi, S., Krawczyk, H.: Strengthening digital signatures via randomized hashing. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 41–59. Springer, Heidelberg (2006)
19. Hong, D., Koo, B., Sasaki, Y.: Improved Preimage Attack for 68-Step HAS-160. In: Lee, D., Hong, S. (eds.) ICISC 2009. LNCS, vol. 5984, pp. 332–348. Springer, Heidelberg (2010)
20. Indesteege, S.: The LANE hash function. Submission to NIST (2008)
21. International Organization for Standardization. ISO/IEC 10118-3:2004, Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions (2004)
22. Jean, J., Naya-Plasencia, M., Peyrin, T.: Improved Rebound Attack on the Finalist Grøstl. In: Canteaut (ed.) [7], pp. 110–126
23. Joux, A. (ed.): FSE 2011. LNCS, vol. 6733. Springer, Heidelberg (2011)
24. Khovratovich, D., Nikolić, I.: Rotational Cryptanalysis of ARX. In: Hong, S., Iwata, T. (eds.) FSE 2010. LNCS, vol. 6147, pp. 333–346. Springer, Heidelberg (2010)
25. Knudsen, L.R., Rijmen, V.: Known-Key Distinguishers for Some Block Ciphers. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 315–324. Springer, Heidelberg (2007)
26. Lamberger, M., Mendel, F., Rechberger, C., Rijmen, V., Schläffer, M.: Rebound Distinguishers: Results on the Full Whirlpool Compression Function. In: Matsui (ed.) [31], pp. 126–143
27. Landelle, F., Peyrin, T.: Cryptanalysis of full RIPEMD-128. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 228–244. Springer, Heidelberg (2013)
28. Leurent, G.: Construction of Differential Characteristics in ARX Designs - Application to Skein. IACR Cryptology ePrint Archive, 2012:668 (2012)

29. Leurent, G., Nguyen, P.Q.: How Risky Is the Random-Oracle Model? In: Halevi (ed.) [17], pp. 445–464
30. Marshall, A.W., Olkin, I., Arnold, B.C.: Inequalities: Theory of Majorization and Its Applications, 2nd edn. Springer (2011)
31. Matsui, M. (ed.): ASIACRYPT 2009. LNCS, vol. 5912. Springer, Heidelberg (2009)
32. Matusiewicz, K., Naya-Plasencia, M., Nikolic, I., Sasaki, Y., Schläffer, M.: Rebound Attack on the Full Lane Compression Function. In: Matsui (ed.) [31], pp. 106–125
33. Mendel, F., Nad, T., Schläffer, M.: Cryptanalysis of Round-Reduced HAS-160. In: Kim, H. (ed.) ICISC 2011. LNCS, vol. 7259, pp. 33–47. Springer, Heidelberg (2012)
34. Mendel, F., Nad, T., Schläffer, M.: Improving local collisions: New attacks on reduced SHA-256. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 262–278. Springer, Heidelberg (2013)
35. Merkle, R.C.: One Way Hash Functions and DES. In: Brassard (ed.) [6], pp. 428–446
36. Naya-Plasencia, M.: How to Improve Rebound Attacks. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 188–205. Springer, Heidelberg (2011)
37. Naya-Plasencia, M., Toz, D., Varici, K.: Rebound Attack on JH42. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 252–269. Springer, Heidelberg (2011)
38. New European Schemes for Signatures, Integrity, and Encryption (NESSIE). NESSIE Project Announces Final Selection of CRYPTO Algorithms (2003), https://www.cosic.esat.kuleuven.be/nessie/deliverables/press_release_feb27.pdf
39. Nikolić, I., Pieprzyk, J., Sokołowski, P., Steinfeld, R.: Known and Chosen Key Differential Distinguishers for Block Ciphers. In: Rhee, K.-H., Nyang, D. (eds.) ICISC 2010. LNCS, vol. 6829, pp. 29–48. Springer, Heidelberg (2011)
40. Peyrin, T.: Improved Differential Attacks for ECHO and Grøstl. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 370–392. Springer, Heidelberg (2010)
41. Preneel, B.: Analysis and design of cryptographic hash functions. PhD thesis (1993)
42. Rijmen, V., Barreto, P.S.L.M.: The WHIRLPOOL Hashing Function. Submitted to NESSIE (September 2000)
43. Rogaway, P.: Formalizing Human Ignorance. In: Nguyên, P.Q. (ed.) VIETCRYPT 2006. LNCS, vol. 4341, pp. 211–228. Springer, Heidelberg (2006)
44. Sasaki, Y.: Meet-in-the-Middle Preimage Attacks on AES Hashing Modes and an Application to Whirlpool. In: Joux (ed.) [23], pp. 378–396
45. Sasaki, Y., Wang, L., Takasaki, Y., Sakiyama, K., Ohta, K.: Boomerang Distinguishers for Full HAS-160 Compression Function. In: Hanaoka, G., Yamauchi, T. (eds.) IWSEC 2012. LNCS, vol. 7631, pp. 156–169. Springer, Heidelberg (2012)
46. Sasaki, Y., Wang, L., Wu, S., Wu, W.: Investigating Fundamental Security Requirements on Whirlpool: Improved Preimage and Collision Attacks. In: Wang, Sako (eds.) [53], pp. 562–579
47. Shoup, V. (ed.): CRYPTO 2005. LNCS, vol. 3621. Springer, Heidelberg (2005)
48. Telecommunications Technology Association. Hash Function Standard Part 2: Hash Function Algorithm Standard, HAS-160 (2000)
49. U.S. Department of Commerce, National Institute of Standards and Technology. Federal Register 72(212), Notices (November 2, 2007), http://csrc.nist.gov/groups/ST/hash/documents/FR_Notice_Nov07.pdf
50. U.S. Department of Commerce, National Institute of Standards and Technology. Randomized Hashing for Digital Signatures (NIST Special Publication 800-106) (February 2009), http://csrc.nist.gov/publications/nistpubs/800-106/NIST-SP-800-106.pdf

51. U.S. Department of Commerce, National Institute of Standards and Technology. Secure Hash Standard (SHS) (Federal Information Processing Standards Publication 180-4) (2012), http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf
52. Wagner, D.: A Generalized Birthday Problem. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 288–303. Springer, Heidelberg (2002)
53. Wang, X., Sako, K. (eds.): ASIACRYPT 2012. LNCS, vol. 7658. Springer, Heidelberg (2012)
54. Wang, X., Yin, Y.L., Yu, H.: Finding Collisions in the Full SHA-1. In: Shoup (ed.) [47], pp. 17–36
55. Wang, X., Yu, H.: How to Break MD5 and Other Hash Functions. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 19–35. Springer, Heidelberg (2005)

Appendix

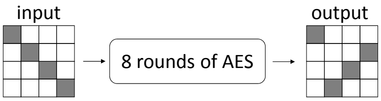


Fig. 2. The limited-birthday distinguisher on AES 8 rounds by Gilbert and Peyrin [15]. Distinguishers aim to find a pair of values satisfying the above truncated differential forms for input and output. Grey cells represent the bytes where any difference is acceptable. Therefore, the number of active bits for the input state is 32 bits, namely, $I = 32$. and similarly, $O = 32$. Inactive bits are represented by empty cells.