Mind the Gap - A Closer Look at the Security of Block Ciphers against Differential Cryptanalysis

Ralph Ankele[1⋆], Stefan Kölbl[2,3]

[1] Royal Holloway University of London, Egham, United Kingdom.
ralph.ankele.2015@rhul.ac.uk
[2] DTU Compute, Technical University of Denmark, Denmark
[3] Cybercrypt, Denmark.
stek@mailbox.org

Abstract. Resistance against differential cryptanalysis is an important design criteria for any modern block cipher and most designs rely on finding some upper bound on probability of single differential characteristics. However, already at EUROCRYPT'91, Lai et al. comprehended that differential cryptanalysis rather uses *differentials* instead of single *characteristics*.

In this paper, we consider exactly the gap between these two approaches and investigate this gap in the context of recent lightweight cryptographic primitives. This shows that for many recent designs like `Midori`, `Skinny` or `Sparx` one has to be careful as bounds from counting the number of active S-boxes only give an inaccurate evaluation of the best differential distinguishers. For several designs we found new differential distinguishers and show how this gap evolves. We found an 8-round differential distinguisher for `Skinny`-64 with a probability of $2^{-56.93}$, while the best single characteristic only suggests a probability of 2^{-72}. Our approach is integrated into publicly available tools and can easily be used when developing new cryptographic primitives.

Moreover, as differential cryptanalysis is critically dependent on the distribution over the keys for the probability of differentials, we provide experiments for some of these new differentials found, in order to confirm that our estimates for the probability are correct. While for `Skinny`-64 the distribution over the keys follows a Poisson distribution, as one would expect, we noticed that `Speck`-64 follows a bimodal distribution, and the distribution of `Midori`-64 suggests a large class of weak keys.

Keywords: Symmetric-key cryptography, differential cryptanalysis, lightweight cryptography, SAT/SMT solver, IoT, `LBlock`, `Midori`, `Present`, `Prince`, `Rectangle`, `Simon`, `Skinny`, `Sparx`, `Speck`, `Twine`

⋆ This research was partially supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. H2020-MSCA-ITN-2014-643161 ECRYPT-NET.

1 Introduction

Differential cryptanalysis, first published by Biham and Shamir [9] to analyse the DES, has become one of the prime attack vectors which any modern symmetric-key primitive has to be resistant against. The idea behind differential cryptanalysis is to find a correlation between the difference of a pair of plaintexts and ciphertexts which holds with high probability. The challenge for an cryptanalyst consists of finding such a correlation or to show that no such correlation exists. A popular approach is to design a cipher in such a way that one can find a bound on the best differential characteristics, either directly e.g., the wide-trail strategy deployed in AES or using methods based on Matsui's algorithm, MILP or SAT.

A differential characteristic specifies all the intermediate differences after each round of the primitive. However, when constructing a differential distinguisher one only cares about the input and output difference. It is often assumed that a single characteristic dominates the probability of such a differential, however this is not true in general and leads to inprecise estimates of the probability in many cases [10,24].
In the work by Lai, Massey and Murphy [33] they showed that if an iterated cryptographic primitive has independent round-keys, it can be considered as a Markov cipher. As differential cryptanalysis considers just the first and last difference and ignores the intermediate values, the probability of such a *differential* can then be computed as the sum of all characteristics, that are formed by the differentials. While this assumes that the rounds are independent, it provides a more precise estimate and the probability of the most probable *differential* will always be greater than the probability of the most probable *characteristic*.

Contributions. We provide a broad study covering different design strategies and investigate the differential gap between single *characteristics* and *differentials* for the block ciphers LBlock, Midori, Present, Prince, Rectangle, Simon, Skinny, Sparx, Speck and Twine. In order to do this, we use an automated approach for enumerating the characteristics with the highest probability contributing to a differential based on SMT solvers [41], which we adopt to different design strategies. This allows us to efficiently enumerate a large set of characteristics contributing to the probability of a differential resulting in a precise estimate for the probability of differentials.

For Skinny-64 we present an 8-round differential distinguisher with a probability of $2^{-56.93}$, while the best single characteristic only suggests a probability of 2^{-72}. For Midori-64 we show that the best characteristic for 8 rounds, with a probability of 2^{-76} can be used to find a differential with a probability of $2^{-60.86}$. Our results show that in the case of many new lightweight ciphers like Midori-64, Skinny-64, and Sparx-64 the probabilities improve significantly and that we can find differential distinguishers which are able to cover more rounds. This suggests that one should be particularly careful with lightweight block ciphers when using simpler approximations like counting the number of active S-boxes.

Our method is generic and can easily be applied to other designs as one only needs to describe the differential behaviour of the round function and can re-use

all the components we implemented for doing so. This allows both to find optimal differential characteristics and to enumerate all characteristics contributing to a differential.

Furthermore, we provide experiments to verify that our estimates of the differential probability provide a good approximation. However, we also noticed that the distribution over the choice of keys varies significantly for some design strategies and that commonly made assumptions do not hold for reduced-round versions. While for Skinny-64 the distribution over the keys follows relatively closely what one would expect we noticed that for Midori-64 for a large class of keys there are no pairs following the differential at all, while for very few keys the probability is significantly higher.

Related Work. Daemen and Rijmen firstly studied the probability of differentials for AES in their work on Plateau Characteristics [20]. In their work, they analysed AES on the distribution of differential probability over the choice of keys and showed that all 2-round characteristics have either a zero probability or for a small subset of keys the probability is non-zero. However, they only considered AES, but conjectured that other ciphers with 4-uniform S-boxes will show a similar result. In the case of AES and AES-like ciphers, there has also been a lot of research in studying the expected differential/linear probability (MEDP/MELP) [18,30], that is used to provable bound the security of a block cipher against differential/linear cryptanalysis.

In recent years, many automated tools were proposed that could help designers to prove bounds against differential/linear attacks. Mouha et al. [42] used Mixed Integer Linear Programming (MILP) to count active S-boxes and compute provable bounds. Furthermore, there have been a few approaches of using automated tools to find optimal characteristics, and to collect many characteristics with the same input/output differences. This idea was first introduced by Sun et al. [46] who used MILP. Likewise, tools using SAT/SMT solvers are used where the results were applied to Salsa-20 [41], Norx [5], and Simon [31].

Moreover, there exist several design and attack papers that study the effect of numerous characteristics contributing to the probability of a differential: Mantis [24], Noekeon [29], Salsa [41], Simon/Speck [11,31], Rectangle [54] and Twine [10]. Yet, these are often based on truncated differentials or dedicated algorithms for finding large numbers of characteristics. For example in [25], Eichlseder and Kales attack Mantis-6 by finding a large cluster of differential characteristics. Contrary to the attack on Mantis-5 by Dobraunig et al. [24] where the cluster was found manually, in the attack on Mantis-6, Eichlseder and Kales used a tool based on truncated differentials.

Similar effects have also been observed in the case of linear cyrptanalysis, where Abdelraheem et al. [1] showed that the security margins based on the distribution of linear biases are not always accurate. Their work has further been studied and improved by Blondeau and Nyberg [13].

Software. All the models for enumerating the differential characteristics are publicly available at `https://github.com/TheBananaMan/cryptosmt`.

Outline. The remainder of this paper is structured as follows. After briefly revisiting some of the necessary definitions about differential cryptanalysis in Section 2, we provide details about the automated tools that we use in Section 3 and describe how to efficiently find differential characteristics for various ciphers. In Section 4 we present the results of our analysis on the gap between single differential characteristics and differentials for various cryptographic primitives. We also analyze the best differential attacks, that are published on those ciphers so far, and show if the attacks can be improved by considering the aforementioned differential gaps. Moreover, in Section 5 we give details about our experiments of the distribution over keys for the probability of differentials.

2 Differentials and Differential Characteristics

Differential cryptanalysis is one of the most powerful techniques in the analysis of symmetric-key primitives. Many extensions to it have been developed and it has found wide applications on block ciphers, stream ciphers and cryptographic hash functions. In the following, we state some definitions and notations that we will use throughout the paper.

A *block cipher* is a family of permutations parameterised by a *key* $K \in \mathbb{F}_2^k$, that maps a set of *plaintexts* $P \in \mathbb{F}_2^n$ to a set of *ciphertexts* $C \in \mathbb{F}_2^n$

$$\mathrm{E}_K : \mathbb{F}_2^k \times \mathbb{F}_2^n \to \mathbb{F}_2^n \,. \tag{1}$$

Virtually all currently used block ciphers are *iterative* block ciphers, i.e., they are composed of applying a simple round function r times

$$\mathrm{E}_K(\cdot) = f_r(\cdot) \circ \ldots \circ f_1(\cdot). \tag{2}$$

The idea of differential cryptanalysis is to look at pairs of plaintexts (p_1, p_2) and the corresponding ciphertexts (c_1, c_2) and try to find a correlation between the differences α and β, where $\alpha = p_1 \oplus p_2$ and $\beta = c_1 \oplus c_2$.

Definition 1. *A* differential *is a pair of differences* $(\alpha, \beta) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$.

If such a correlation holds with high probability, we can use this to distinguish the block cipher from a random permutation and further use this to mount key-recovery attacks.

Definition 2. *The* differential probability *of a differential over a block cipher is*

$$\mathrm{DP}(\alpha \xrightarrow{E_K} \beta) = \Pr_X(E_K(X) \oplus E_K(X \oplus \alpha) = \beta). \tag{3}$$

where X is a random variable that is uniformly distributed over $\mathbb{F}_2^n$.

For ease of notation we define the *weight* of a differential as $-\log_2(\mathrm{DP}(\cdot))$. Any non-zero differential for a random permutation $F_\$: \mathbb{F}_2^n \to \mathbb{F}_2^n$ will have a differential probability close to 2^{-n}. Therefore one is interested in finding any differential with $\mathrm{DP}(\alpha \xrightarrow{\mathrm{E}_K} \beta) \gg 2^{-n}$. In general, it is computationally infeasible to compute the exact value of the DP as this would require to exhaustively search through the whole space of all possible plaintexts. One can use the structure of a block cipher, to obtain a good approximation of the actual DP with less computational effort by tracking the differences through the round functions.

Definition 3. *A* differential characteristic *is a sequence of differences*

$$Q = (\alpha_1 \xrightarrow{f_1} \alpha_2 \xrightarrow{f_2} \ldots \xrightarrow{f_{r-1}} \alpha_r). \tag{4}$$

Yet, it is still computationally infeasible to compute the exact value of $\mathrm{DP}(Q)$ and the general approach is to assume independence of the rounds. For most designs it is feasible to compute the exact probability of a differential for a single round. One can therefore compute

$$\mathrm{DP}(Q) \approx \prod_{i=1}^{r-1} \Pr_X(\alpha_i \xrightarrow[X]{f_i} \alpha_{i+1}). \tag{5}$$

While this assumption of independent rounds is not true in general, it has been shown to serve as a good approximation in practice. However, if an adversary wants to construct a distinguisher, she actually does not care about any intermediate differences and is only interested in the probability of the differential. The adversary can therefore collect all differential characteristics sharing the same input and output difference to get a better estimate

$$\Pr(\alpha_1 \xrightarrow{E} \alpha_r) = \sum_{\alpha_2,\ldots,\alpha_{r-1}} \Pr_X(\alpha_1 \xrightarrow[X]{f_1} \alpha_2 \xrightarrow[f_1(X)]{f_2} \cdots \alpha_{r-1} \xrightarrow[f_{r-1}\circ\ldots\circ f_1(X)]{f_{r-1}} \alpha_r). \tag{6}$$

It is often assumed that the probability of the differential is close to the probability of the best single characteristic. While this might hold for some ciphers this assumption has been shown to be inaccurate in several cases and does not hold for many modern block ciphers [10,24]. We will show later in Section 4 that this assumption fails particularly often for some recently designed lightweight block ciphers.

We consider two different criteria for a design: *differential characteristic resistant* (DCR), which means that no single differential characteristic exists with a probability larger than 2^{-n} and *differential resistant* (DR) which means that it should be difficult to find a differential with a probability larger than 2^{-n}. Note that we typically can not avoid that there are differentials with $\mathrm{DP} \geq 2^{-n}$, as if we fix the input difference to α_1 then $\sum_{\alpha_r \neq 0} \Pr(\alpha_1 \xrightarrow{E} \alpha_r) = 1$. This implies that there exists at least one differential with a probability $\mathrm{DP} \geq 2^{-n}$. In the *Wide-Trail Strategy* which was used to design the `AES` and subsequently many other ciphers, Daemen and Rijmen suggest that it is a sound design strategy to

restrict the probability of difference propagation [19]. Nevertheless, this does not result in a proof for security.

Note that in the definitions so far the influence of the keys was ignored. However, the DP for a specific differential strongly depends on the choice of the secret key and it is therefore of interest how this distribution looks like. To solve this problem we could compute the probabilities of a differential over the whole key space, however this is again practically infeasible which leads one to use the *expected differential probability*.

Definition 4. *The* expected differential probability *of a block cipher E_k of an r-round differential (α, β), with a key-size of κ-bits is defined as*

$$\mathrm{EDP}(\alpha \xrightarrow{E} \beta) = 2^{\kappa} \sum_{k \in \mathbb{F}_2^{\kappa}} \Pr_X(\alpha \xrightarrow[X]{E_k} \beta). \tag{7}$$

In order to derive some sort of security proof against differential cryptanalysis often the *Hypothesis of Stochastic Equivalence* [33] is used which states that for all differentials Q it holds that for most keys K the differential probability of a characteristic is close to the expected differential probability, $\mathrm{DP}_K(Q) \approx \mathrm{EDP}(Q)$. In practice this hypothesis does not always hold [16], which we will also see later in Section 5.

3 Finding Differential Characteristics Efficiently

While there are many methods based on SAT, MILP or Matsui's algorithm to find differential characteristics and even prove an upper bound on the probability of the best single characteristic, it remains a hard problem to find a good estimate on the probability of the best differential. Even finding those differential characteristics remains a difficult problem for some design strategies and cryptanalysts had to search manually for differentials in some attacks [53]. Nowadays a variety of automated tools [45,12,35] is available which are constantly improved and help cryptanalysts in finding good differential characteristics.

3.1 SAT/SMT Solvers

SAT solvers are used to solve the Boolean satisfiability problem (SAT) and are based on heuristic algorithms. A solver starts from an initial assignment for the literals and then builds a search tree by using systematic backtracking until all conflicting clauses are resolved and either an assignment of variables for a satisfiable set of clauses is returned or the solver decides that this instance is unsatisfiable. The most commonly algorithms used in SAT solvers are based on the original idea of DPLL [21].

SMT solvers are more powerful than SAT solvers in the sense that they can express constraints on a higher abstraction layer and allow simple first-order logic. In general, SMT solvers often translate the problem to SAT and then use an improved version of the DPLL algorithm and backtracking to infer when theory

conflicts arise. Moreover, the solver checks the feasibility of conjunctions from the first-order logic predicates as it interacts with the Boolean formulas that are returned by the SAT solver.

There exists a few SAT/SMT solvers that are suitable for our use cases. STP [50] is an SMT solver that uses the CVC and SMTLIB2 language to encode the constraints and then invokes a SAT solver to check for satisfiability of the model. CryptoMiniSat [40] is an advanced SAT solver that supports features like XOR recovery[4] to simplify clauses. As XOR operations are commonly used in cryptography this can be an advantage and potentially reduces the solving time. We also considered other solvers like Boolector [43], which for some instances provide a better performance, however in general this only provides an improvement by a small constant factor and it is hard to identify for which instances one obtains any advantage.

3.2 From Differential Cryptanalysis to Satisfiability Modulo Theories

When using automated tool like SAT/SMT solvers, one can simplify the search for differential characteristics and differentials by modeling the differential behavior of the block cipher. For this we represent all intermediate states of our block cipher as variables which corresponds to the differences and encode the transitions of differences through the round functions as constraints that can be processed by the SMT/SAT solver. An advantage of using SMT over SAT for the modeling is that most SMT solvers support reasoning over bit-vectors which are commonly used in block cipher designs, especially when considering word-oriented ciphers. This both simplifies the modeling of the constraints and can lead to an improved time for solving the given problem instances compared to an encoding in SAT.

Constructing an SMT Model. In this paper, we focus on a tool that uses the CVC language[5] for encoding the differential behavior of block ciphers. Therefore, we encode the constraints imposed by the round function for each round of the block cipher and the probability of the resulting differential transitions. Our main goal here is to construct an SMT model which decides whether

$$\exists Q : \mathrm{DP}(Q) = 2^{-t}, \tag{8}$$

which allows us to find the best differential characteristic Q for a cipher by finding the minimum value t for which the model is satisfiable.

In order to represent the differential behaviour of a cipher we consider any operation in the cipher, e.g., the application of an S-box, matrix multiplication, word-wise operation or bit operation, and add constraints for a valid transition from an input to an output difference such that any valid assignment to the

[4] See `https://www.msoos.org/2011/03/recovering-xors-from-a-cnf/`

[5] A list of all bitwise and word level functions in CVC is available at: `http://stp.github.io/cvc-input-language/`

variables corresponds to a valid differential characteristic in the actual operation. For any non-linear component we introduce additional variables w^j which represent the $\log_2$ probability of the differential transition. The probability of Q is then given by $\sum w^j$. This means that a valid assignment for all these variables directly gives us the differential characteristic Q with all intermediate differences and $\mathrm{DP}(Q) = p$.

In the following we give an overview on how the different components of the ciphers can be modeled in the SMT model. The algorithms to find the optimal differential characteristics and consequently good estimates for the differentials are described in Section 3.3.

S-boxes. Substitution Permutation Network (SPN) ciphers typically use S-boxes, which are non-linear functions operating on a small number of bits. These are often 4- or 8-bit functions and therefore we can compute the differential probability by simply constructing the Difference Distribution Table (DDT), which is a full lookup table of all possible pairs of input/output differences, for each S-box. In our SMT model we represent the input difference to an n-bit S-box as $\alpha = \alpha_1, \ldots, \alpha_n$ respectively the output as $\beta = \beta_1, \ldots, \beta_n$. These variables correspond to the input/output difference to this S-box and we want to constraint them to only allow non-zero probability combinations of input and output differences. We further introduce additional variables $w = w_1, \ldots, w_n$ which are used to represent the probability of the transition. The probability of the transition is encoded as $2^{-wt(w)}$, where $wt(\cdot)$ denotes the Hamming weight of w.

In order to construct the constraints on the variables, we first find all valid transitions and their corresponding probability. We want to construct a CNF which is satisfiable if and only if the assignment corresponds to such a valid characteristic. One simple way to this is by just considering all assignments which are impossible. If a transition is defined as $(a \xrightarrow{S} b)$ and has a probability c then we add the following clause

$$\begin{aligned} T = \; & N(a_1, \alpha_1) \vee \ldots \vee N(a_n, \alpha_n) \vee \\ & N(b_1, \beta_1) \vee \ldots \vee N(b_1, \beta_n) \vee \\ & N(c_1, w_1) \vee \ldots \vee N(c_n, w_n) \end{aligned} \tag{9}$$

where

$$N(x_i, y_i) = \begin{cases} \neg y_i, \text{if } x_i = 0 \\ y_i, \text{if } x_i = 1 \end{cases}. \tag{10}$$

This clause is only satisfiable if the variables of the corresponding S-box are not set to the invalid assignment. For example let $a = (1, 0, 1, 1)$, $b = (0, 0, 0, 0)$ and $c = (0, 0, 0, 0)$ then we add the clause

$$(\neg\alpha_0 \vee \alpha_1 \vee \neg\alpha_2 \vee \neg\alpha_3 \vee \beta_0 \vee \beta_1 \vee \beta_2 \vee \beta_3 \vee w_0 \vee w_1 \vee w_2 \vee w_3). \tag{11}$$

We implemented this approach to generate the SMT models for 4- and 8-bit S-boxes, where most of the lightweight ciphers actually use 4-bit S-boxes which

allows a very compact description (i.e., to represent the 4-bit S-box of `Skinny` we need 12 variables and about 3999 clauses in CNF). Note that our method is limited to S-boxes which have a DDT with entries that are a power of 2. For other S-boxes a similar method could be used by using l additional variables for encoding probabilities of the form $2^{-0.5}, 2^{-0.25}, \ldots$ to get an approximation of the actual probability.

Linear Layers. The diffusion layers of Substitution Permutation Networks in lightweight ciphers are often constructed with simple bit-permutations (e.g., `Present`) or by multiplication with matrices having only binary coefficients (e.g., `Midori`, `Skinny`). ARX-based ciphers (e.g., `Speck`) use the diffusion properties of XOR combined with rotations. Feistel networks (e.g., `Simon`, `LBlock`, `Twine`) also mix the state by switching parts of the states on every Feistel switch.

For modeling rotations and bit-permutations in an SAT/SMT solver, we simply have to re-index the variables accordingly before they are input to another function. This can be achieved using SMT predicates (`ASSERT` and equality) in the CVC language. Rotations can be realized using predicates for *shifting* words and the word-wise *or* function that are available in the CVC language. The multiplication by a binary matrix can be modeled using the *xor* predicate at the word-level.

ARX Designs. ARX designs use modular additions (modulo 2^n), XOR and rotations. As modular addition is the only non-linear component, that is not already available in the SMT solver, we use an algorithm proposed by Lipmaa and Moriai [36] to efficiently compute the differential probability of modular addition. Let $xdp^+(\alpha, \beta \rightarrow \gamma)$ be the XOR differential probability of modular addition, where α, β are input differences and γ is the output difference, then it holds that a differential is valid if and only if:

$$\mathrm{eq}(\alpha \ll 1, \beta \ll 1, \gamma \ll 1) \wedge (\alpha \oplus \beta \oplus \gamma \oplus (\beta \ll 1)) = 0 \tag{12}$$

where

$$\mathrm{eq}(x, y, z) := (\neg x \oplus y) \wedge (\neg x \oplus z). \tag{13}$$

The weight of a valid differential is defined as:

$$w(\alpha, \beta, \gamma) := -\log_2 \left(xdp^+(\alpha, \beta \rightarrow \gamma)\right) = wt'(\neg eq(\alpha, \beta, \gamma)). \tag{14}$$

where $wt'(\cdot)$ denotes the Hamming weight omitting the most significant bit. We implemented this algorithm to calculate the differential probability of modular additions.

3.3 Finding the Best Characteristics and Differentials

We use the open-source tool CryptoSMT [45] for the automated search of differential characteristics and implemented several missing functionalities for block

ciphers (i.e., support for S-boxes as described in Section 3.2, and binary diffusion matrices). CryptoSMT is based on the state-of-the-art SAT/SMT solvers, CryptoMiniSat [40] and STP [50].

The tool offers a simple API that allows cryptanalysts and designers to formulate various cryptanalytic problems and solve them with the underlying SAT/SMT solver. We added the models for the block ciphers `Skinny`, `Midori`, `Rectangle`, `Present`, `Prince`, `Sparx`, `Twine` and `LBlock` (note that some of these are block cipher families and we focused on a subset of parameters) to CryptoSMT and use the following two functionalities provided by the tool:

- Decide if a differential characteristic with probability p exists.
- Enumerate all differential characteristics with a probability of p.

Based on this we can achieve our two goals, namely finding the best differential characteristic and estimating the probability of the differential.

Best Differential Characteristic. In order to find the characteristic Q with maximum probability $p_{\max}$ for r rounds of a block cipher we start by checking whether our model is satisfiable for a probability of p, starting at $p = 1$. If our model is not satisfiable we continue by checking whether there is a valid assignment for $p = 2^{-1}$. Note that for all our block ciphers the probability of the differential transitions are powers of two and therefore there does not exist any differential characteristic which has a probability p' such that $2^{-(t+1)} < p' < 2^{-t}$ for any integer t. We continue this process until we reach a model which is satisfiable, which gives us an assignment of all variables of the state forming a valid differential characteristic with probability $p_{\max} = 2^{-t}$. Considering that we start with probability $p = 1$ and then we constantly increase the weight, and finish as soon as we found an valid assignment, we can ensure that we found the best differential characteristic.

Estimate for the Probability of a Differential. In order to find a good differential we can use a tool assisted approach to compute an approximation for Equation 6, as shown in [41]. We first obtain the best single characteristic Q with probability $p = 2^{-t}$ which gives us the input difference α_1 and output difference α_r. Subsequently we modify our model and fix the input and output difference to α_1 respectively α_r. Note that this restricts the search space significantly and results in a much faster time for solving any subsequent SMT instances.

The next step is to find all differential characteristics Q, such that $\mathrm{DP}(Q) = 2^{-u}$, for $u = t, t+1, \ldots$, under this new constraints. This allows us to collect more and more terms of the sum in Equation 6, improving our approximation for the differential. By doing this process we always search for those differential characteristics which contribute the most to the probability of the differential first.

Here we assume that the input and output difference imposed by the best differential characteristic correspond to a good differential. While this assumption might not always hold and some of the differentials we found significantly improve

Table 1. Best attacks and security margins (active S-boxes) for various design strategies for symmetric cryptographic primitives. D/MD/RK/ID/R/TD = differential, multiple differential, related-key, impossible differential, rectangle, truncated differential

Group	Design Strategy	Cipher	Block Size	Key Size	Rounds	Margin (active S-boxes)	Best Differential Attack	Exploit Differentials
SPN	AES-like	Midori	64	128	16	9 rounds	full rounds (RK) [26]	✗
		Skinny	64	64	32	24 rounds	19 rounds (ID) [38]	✓
		Skinny	64	128	36	28 rounds	23 rounds (ID) [3,38]	✓
		Skinny	64	192	40	32 rounds	27 rounds (R) [38]	✓
	Bit-sliced	Rectangle	64	80/128	25	-	18 rounds (D) [54,48]	Sec.4.6
	Present-like	Present	64	80/128	31	12 rounds	26 rounds (D) [37,51]	✓
	Reflection	Prince	64	128	12	-	10 rounds (MD) [17]	✓
	ARX	Sparx	64	128	24	9 rounds	16 rounds (TD) [4]	✓
Feistel	AND-RX	Simon	64	96	42	-	26 rounds (D) [2]	✓
		Simon	64	128	44	-	26 rounds (D) [2]	✓
	ARX	Speck	64	96	26	-	19 rounds (D) [44]	✓
		Speck	64	128	27	-	20 rounds (D) [44]	✓
	GFN	Twine	64	80	36	21 rounds	23 rounds (ID) [10]	✗
		Twine	64	128	36	21 rounds	25 rounds (ID) [10]	✗
	Two-branched	LBlock	64	80	32	17 rounds	24 rounds (ID) [52]	✗

the best differential distinguishers there could still exist better starting points for our search, for example as shown in [32] against the block cipher Simeck.

4 Analysis of the Gap in Lightweight Ciphers

The construction of cryptographic primitives optimized for resource constrained devices has received a lot of attention over the last decade and various design strategies and optimisation targets have been explored. All these primitives exhibit the idea of using simpler operations in order to save costs and therefore often exhibit a simpler algebraic structure compared to other symmetric-key algorithms.

For some design strategies this leads to a significant larger gap between single characteristics and differentials. This gap becomes especially relevant for aggressively optimised designs with minor security margins. Table 1 gives an overview of all the block ciphers we analysed with the methodology outlined in Section 3 and their security margins as well as the best known differential attacks.

4.1 Designs Strategies

We categorise these lightweight ciphers according to their design strategies as this has the largest influence on the gap. In general one can distinguish

between two main design families: Substitution-Permutation Networks (SPN) and Feistel Networks. Within these families we can gather ciphers according to other structural properties. These are for SPN: AES-like, Bit-sliced S-boxes, Bit-based Permutation Layers, Reflection Ciphers, ARX-based and for Feistel: ARX-based, Generalized Feistel Networks and Two-branched.

In our study, we then analyzed the differential gaps for Midori [6], Skinny [8], Rectangle [54], Present [14], Prince [15], Sparx [23], Simon [7], Speck [7], Twine [47], and LBlock [47] where Table 1 categorises the ciphers according their aforementioned structural properties.

4.2 Skinny

Skinny [8] is an AES-like tweakable block cipher, based on the Tweakey framework [28]. The aim of Skinny is to achieve the hardware performance of the AND-RX-cipher Simon and have strong security bounds against differential/linear attacks (this includes the related-key scenario), while also having competitive software performance. The resistance against differential/linear attacks in Skinny is based on counting the minimal number of active S-boxes, in the single-key and related-tweakey models. As the design of Skinny is based on a few very simple but highly efficient cryptographic building blocks it seems intuitive that one can expect that a large number of differential characteristics will contribute to a differential. Recent attacks [3,38] exploited the low branch number of the binary diffusion matrix, as well as properties of the tweakey schedule.

Using our tool-assisted approach we analysed this gap in Skinny-64 (see Figure 1) and can provide some new insights to the security of Skinny-64. For example the best 8-round single differential characteristic $Q^8_{\max}$ suggests a probability of 2^{-72} while the differential D^8 defined by the input/output difference of $Q^8_{\max}$ consists of a large cluster of characteristics leading to the differential

$$\texttt{0x0104401000C01C00} \xrightarrow{8-\text{round Skinny-64}} \texttt{0x0606060000060666} \tag{15}$$

with a probability larger than $2^{-56.55}$ by taking all 821896 characteristics[6] into account which have DP $> 2^{-99}$. Note that the probabilities and the number of characteristics are obtained with a fixed input/output difference as noted in Equation 15. This suggests that estimates from active S-boxes should be taken with care as the gap is fairly large. However, the number of rounds in Skinny-64 is chosen very conservatively and it provides a large security margin.

In particular the probability of the differential improves very quickly when adding more characteristics, as the distribution of the number of characteristics with a probability 2^{-t} is very flat over the choice of t (see Figure 1). For example there are 39699 characteristics with DP $= 2^{-75}$ and 25413 characteristics with DP $= 2^{-76}$ and the probability of the differential only improves marginally by considering more characteristics with a lower probability. On the contrary,

[6] This process took in total 23.5 hours on a single core, however after 1 hour the estimate for the differential probability improves by less than $2^{-0.9}$.

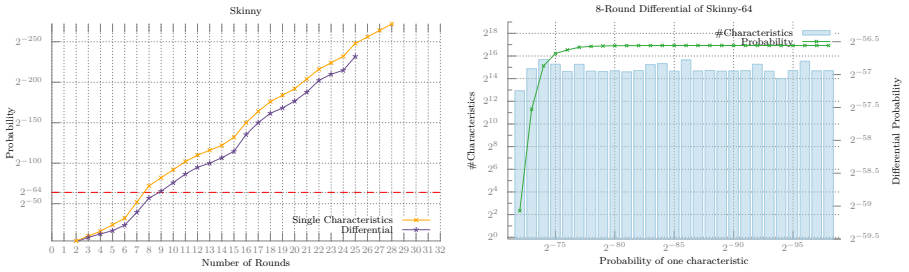

Fig. 1. Probability for the best single characteristics and differentials for `Skinny`-64 (left), and the distribution of the number of characteristics with a fixed probability contributing to the best 8-round differential for `Skinny`-64 (right). The green line indicates the probability of the differential when summing up the probability of all characteristics up to this probability, which highlights the small improvement when adding all lower probability characteristics.

for designs like `Simon` (see Figure 5) this distribution grows exponentially as the probability of the single characteristics decreases as has also been noted in [31], and one has to take a much larger number of characteristics into account before getting a good approximation. For a detailed overview over how many characteristics contribute to each differential see Appendix A.

4.3 `Midori`

`Midori` is an `AES`-like lightweight block cipher optimized for low-energy usage using a binary near-MDS matrix combined with a generic cell permutation for diffusion. Despite that `Midori`-64 has a large number of 2^{32} weak keys, for which `Midori`-64 can be practically broken with invariant subspace attacks [27], there has been no differential attacks on even reduced versions of `Midori`, apart from a related-key attack by Gérault and Lafourcade [26].

The gap between the differential probability of a single characteristic and a differential behaves similar to `Skinny`-64, i.e., counting the active S-boxes gives an inaccurate bound against differential distinguishers. For example we found new differentials for `Midori`-64 where the 8-round single differential characteristic suggests a probability of 2^{-76} and the corresponding 8-round differential

$$\texttt{0x0A000000A0000005} \xrightarrow{8-\text{round } \texttt{Midori-64}} \texttt{0x000000000000A0AA} \tag{16}$$

has a probability larger than $2^{-60.86}$ by summing all 693730 characteristics up to a probability of 2^{-114}. Similar to `Skinny` the distribution of the contributing characteristics is very flat, which means that we quickly approach a good estimate for the probability of the differential (see Figure 2).

4.4 `Sparx`

`Sparx` [23] is based on the *long-trail strategy*, introduced alongside with `Sparx`, which can be seen as combining the ARX approach with an SPN, allowing to

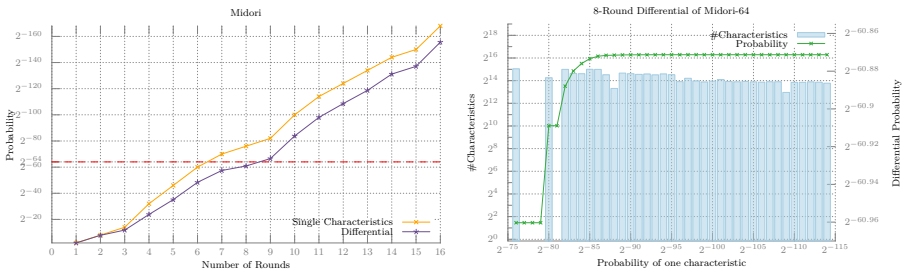

Fig. 2. Probability for the best single characteristics and differentials for various rounds of `Midori`-64 (left), and distribution of the characteristics contributing to the best 8-round differential for `Midori`-64 (right).

provide bounds on the differential resistance of an ARX cipher by counting the active S-boxes. While it is also feasible to prove such a bound using the methodology from Section 3, it is often computationally infeasible or the bounds are not very tight [41]. The designers of `Sparx` used the YAARX toolkit [12] to show truncated characteristics, that they used to compute the differential bounds. One of the main design motivations of `Sparx` was that it should be very difficult to find differential characteristics for a large number of rounds for ARX-based ciphers with a state of more than 32 bits [22].

In general ARX ciphers do not have a very strong differential effect compared to the previous lightweight SPN constructions, however as `Sparx` is in-between those it is an interesting target. Our results suggest that `Sparx`-64 has a differential effect comparable to other ARX designs like `Speck`-64 (see Figure 3). The major limitation for applying our approach to `Sparx` is that the search for optimal differential characteristics on `Sparx` is computationally very costly. While single-characteristics up to 6 rounds can be found in less then 5 minutes, the 10-round single-characteristic took already 32 days, on a single core[7].

4.5 Results for other Lightweight Ciphers

Table 2 summarizes the gaps between single-characteristics and differentials for all lightweight block ciphers we analyzed. We observed that for most ciphers a large gap between the probability for single-characteristics and differentials exists and that a higher number of rounds is required for the block ciphers to be *differential resistant*. The gaps also increase significantly with the number of rounds, which is not surprising as with more rounds there are more valid differential characteristics for a given input/output difference.

The biggest gap, in term of number of rounds, occurs for `Simon`-64 with a gap of five rounds. There is also a 2-round gap for ciphers like `Present`, `Midori` and `Twine`. However, it seems that the gap for `Simon`-64 grows faster, considering

[7] Note that this process can not easily be parallelized as most SAT solvers are inherently serial.

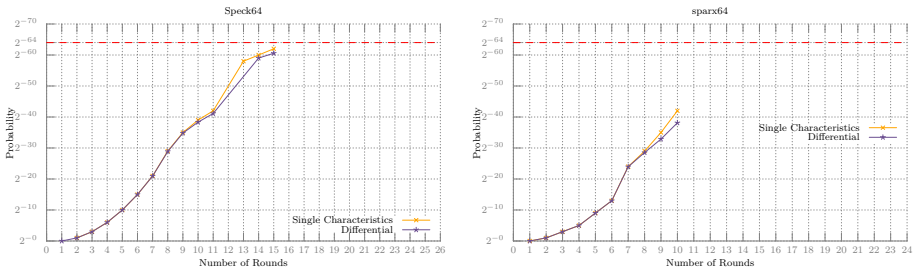

Fig. 3. Comparison of the best single characteristics and differentials for various rounds of `Speck`-64 (left), and `Sparx`-64 (right).

Table 2. Gap between the number of rounds required for a cipher to be *differential characteristic resistant* (DCR) and *differential resistant* (DR). Note that DR is only a lower bound and there might still exist better differentials.

Group	Design Strategy	Cipher	Block Size	Key Size	Rounds	DCR	DR
SPN	`AES`-like	`Midori`	64	128	16	7	9
		`Skinny`	64	64/128/192	32	8	9
	Bit-sliced	`Rectangle`	64	80/128	25	15	15
	`Present`-like	`Present`	64	80/128	31	15	17
	Reflection	`Prince`	64	128	12	6	8
	ARX-based	`Sparx`	64	128	24	15	15[8]
Feistel	AND-RX	`Simon`	64	96/128	42	19	24[9]
	ARX	`Speck`	64	96/128	26	> 15	> 15[10]
	GFN	`Twine`	64	80/128	36	14	16
	Two-branched	`LBlock`	64	80	32	15	16

that the differentials and characteristics seem to follow an exponential growth as also observed in [31]. In comparison `Present`, `Midori` and `Twine` seem to grow in a linear way. In relation to the number of rounds, the gap for `Midori` also has quite a significant impact and allows to extend the distinguisher by two rounds. Further we observed that there seem to be nearly no gaps for ciphers like `Rectangle` and `Speck`. We illustrate the gaps for the analyzed ciphers in Figure 4 and we provide Figure 5 showing the distribution of valid differential characteristics that contribute to the probability of the best differential for each cipher.

[8] Single-characteristic differentials of `Sparx` [23] are proven to reach 15 rounds, while the authors mention that they don't expect the bound to be tight.
[9] The best differentials for `Simon`-64 reach 23 rounds with $2^{-63.91}$ [39].
[10] The best differentials for `Speck`-64 reach 15 rounds with $2^{-60.56}$ [44].

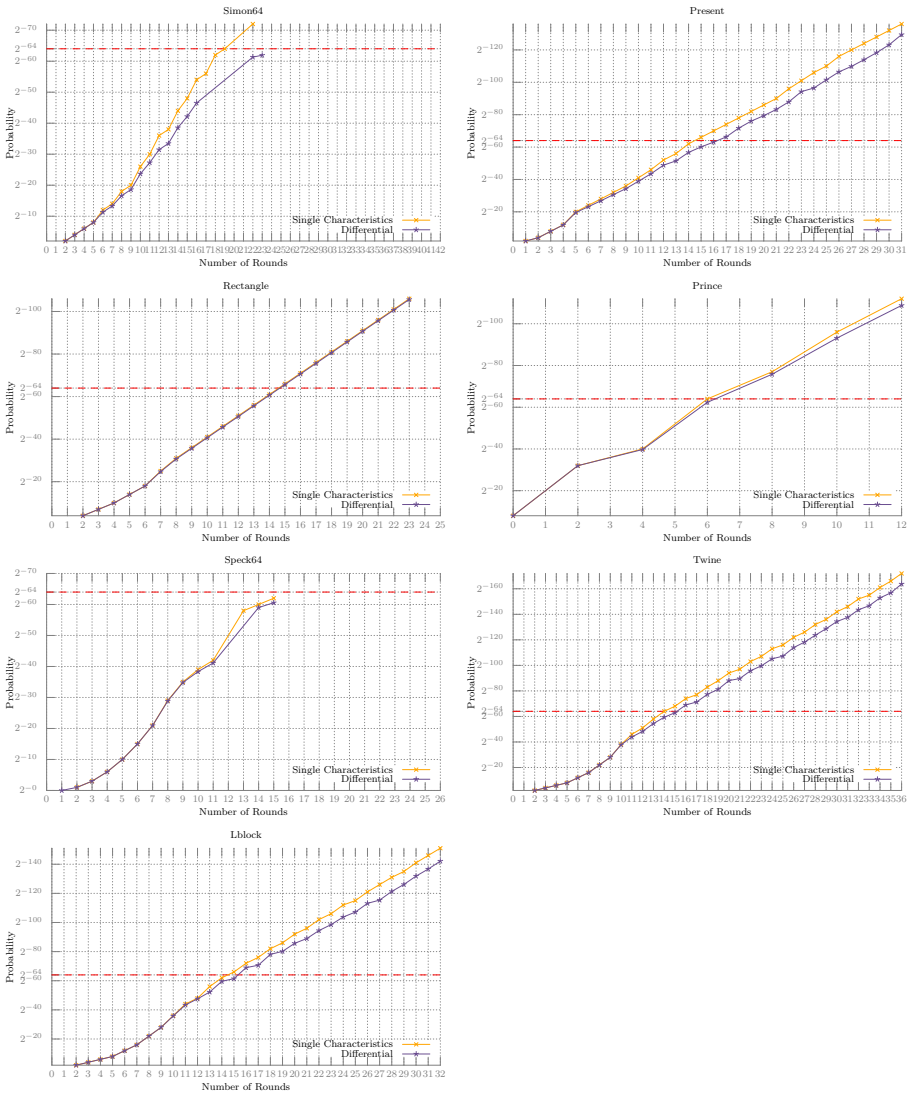

Fig. 4. Probability for the best single characteristics and differentials for various rounds of different block ciphers. 1st row: `Simon`-64 (left) and `Present` (right), 2nd row: `Rectangle` (left) and `Prince` (right), 3rd row: `Speck`-64 (left) and `Twine` (right), 4th row: `LBlock` (left)

4.6 Application of the Differential Gaps to the Best Published Differential Attacks

In the following, we analyze the best published attacks and discuss improvements of the attacks when possible:

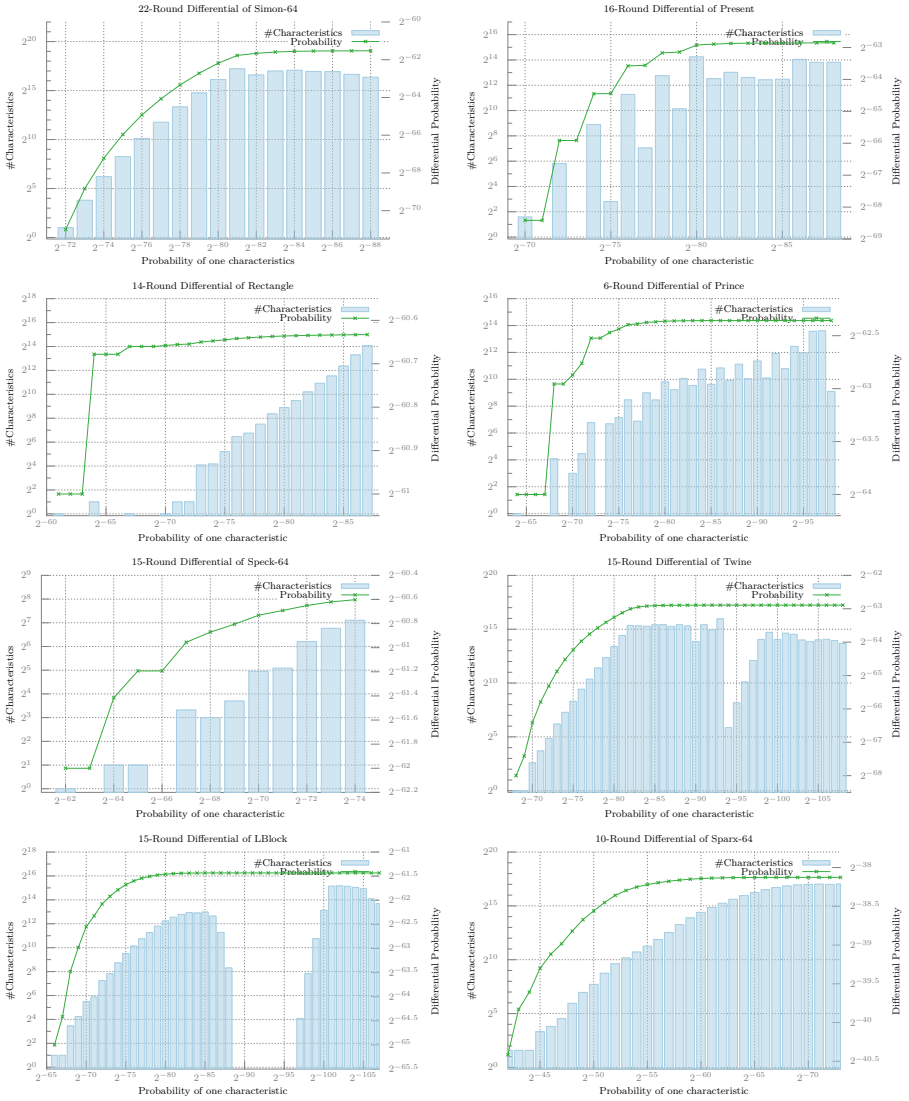

Fig. 5. Distribution of the characteristics contributing to the best differential for various block ciphers. 1st row: `Simon`-64 (left) and `Present` (right), 2nd row: `Rectangle` (left) and `Prince` (right), 3rd row: `Speck`-64 (left) and `Twine` (right), 4th row: `LBlock` (left) and `Sparx`-64 (right)

`Midori`-64. Gérault and Lafourcade [26] proposed related-key differential attacks on full-round `Midori`-64, where they use 16 15-round and $4 \cdot 14$-round related-key differential characteristics to recover the key. In their attacks they do not exploit

differentials. In comparison, the best differential that we found reaches 8 rounds with a probability of $2^{-60.86}$.

Skinny-64. Liu et al. [38] propose related-tweakey rectangle attacks on 26 rounds of Skinny-64-192 and they use optimal single differential characteristics based on truncated differential characteristics. The authors exploit the differential gap of Skinny by using 5000 single differential characteristics to compute the differential for a 22-round distinguisher. In comparison, the best differential characteristic with no differences in the tweak/key that we found reaches 8 rounds with a probability of $2^{-56.55}$.

Rectangle. Zhang et al. [54] studied the differential effect and showed an 18-round differential attack, where they used a 14-round differential with a probability of $2^{-62.83}$. In our analysis we found a better differential for 14 rounds with probability of $2^{-60.63}$ by summing up 40627 single-characteristics which would improve the complexity of these attacks. For more rounds the distinguisher are below 2^{-64}.

Present. Liu and Jin [37] presented an 18-round attack based on slender-sets. Wang et al. [51] further presented normal differential attacks on 16-round Present where they used a differential with probability $2^{-62.13}$ by summing up 91 differential characteristics which is comparable to our differentials.

Prince. Canteaut et al. [17] showed differential attacks on 10 rounds of Prince, by considering multiple differential characteristics. In their attack they use 12 differentials for 6 rounds with a probability of $2^{-56.42}$ by summing up 1536 single-characteristics. The differential we found for 6 rounds only has a probability of about 2^{-62}, but does not lead to further improvements of the attack.

Sparx-64. Ankele and List [4] studied truncated differential attacks on 16 rounds of Sparx-64/128 and used single differential characteristics, for the first part of the 14-round distinguisher, and truncated the second part of the distinguisher. The designers of Sparx-64 claim that Sparx is differential secure for 15 rounds, however, by considering the differential effect of Sparx-64, also in comparison with Speck-64, it seems likely that there exist differentials with more than 15 rounds with a data complexity below using the full codebook.

Simon-64. Abed et al. [2] presented differential attacks on Simon-64, where they used a 21-round distinguisher with a probability of $2^{-61.01}$. Better distinguishers are reported by [39] for 23 rounds with a probability of $2^{-63.91}$. The differentials we found are in line with previous results.

Speck-64. Song et al. [44] presented 20-round attacks on Speck-64 by constructing a distinguisher from two short characteristics where they concatenated the two characteristics to a 15-round characteristic with probability $2^{-60.56}$. The distinguishers used in the attack are already based on differentials and the differentials we found do not lead to any improvement.

Twine. Biryukov et al. [10] showed a 25-round impossible differential attack and a truncated differential attack on 23 rounds by chaining several iterated 4-round characteristics together. In the paper the authors also considered differentials for 12 rounds with a probability of $2^{-52.08}$ and 16 rounds with probability $2^{-67.59}$. The best differential that we found reaches 15 rounds with a probability of $2^{-62.89}$.

LBlock. Wang et al. [52] published a 24-round impossible differential attack on LBlock. Due to the nature of impossible differential attacks, characteristics with probability 1 are used for constructing these. The best differential that we found reaches 15 rounds with a probability of $2^{-61.43}$.

5 Experimental Verification and the Influence of Keys

In Section 2 we made several assumptions in order to compute $\mathrm{DP}(Q)$ and in this section we compare the theoretical estimates with experiments for reduced-round versions. This serves two purpose: First we want to see how close our estimate for $\mathrm{DP}(\alpha, \beta)$ is and secondly we want to see the distribution over the choice of keys. Specifically, we are interested in the number of pairs

$$\delta_K(\alpha, \beta) = \#\{x \in \mathbb{F}_2^n \mid \mathrm{E}_K(x) \oplus \mathrm{E}_K(x \oplus \alpha) = \beta\}. \tag{17}$$

This number of *good* pairs will vary over the choice of the key. For a random process we would expect that the number of valid pairs is about $\mathrm{DP} \cdot 2^n$ and follows a Poisson distribution.

Definition 5. *Let X be a Poisson distributed random variable representing the number of pairs (a, b) with values in $\mathbb{F}_2^n$ following a differential $D = (\alpha \xrightarrow{f} \beta)$, that means $f(a) \oplus f(a \oplus \alpha) = \beta$, then*

$$\Pr(X = l) = \frac{1}{2}(2^n p)^l \frac{e^{-(2^n p)}}{l!} \tag{18}$$

where p is the probability of the differential.

In the following, we experimentally verify differentials for Skinny, Speck and Midori for a large number of random pairs of plaintexts and a random choice of keys to see how good this approximation is.

5.1 Skinny

As a first example we look at Skinny-64. We use the 6-round differential

$$D = (\texttt{0x0000010010000041}, \texttt{0x4444004040044044})$$

for Skinny-64. The best characteristic which is part of D has a probability of 2^{-32} and by collecting all characteristics (100319) contributing to this differential

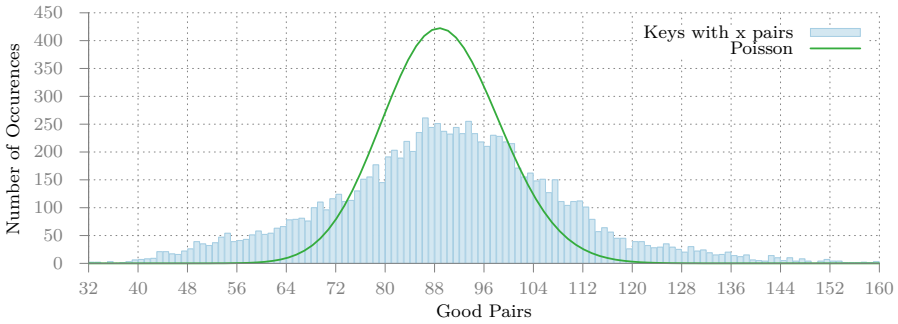

Fig. 6. Distribution of $\delta_K(D)$ over a random choice of K for 6-round `Skinny`-64.

we estimate $\mathrm{DP}(D) \approx 2^{-23.52}$. We try out 2^{30} randomly selected pairs for 10000 keys and count the number of pairs following D. From our estimate we would expect that on average we get about 89 pairs for a key.
As one can see from Figure 6 our estimate of $\mathrm{DP}(D)$ provides a good approximation for the distribution over the keys, although the distribution has a larger variance than we expected.

5.2 `Speck`

For `Speck`-64 we look at the differential

$$D = ((\texttt{0x40004092}, \texttt{0x10420040}), (\texttt{0x8080A080}, \texttt{0x8481A4A0}))$$

over 7 rounds. The best characteristic in D has a probability of 2^{-21} and this only slightly improves to about $2^{-20.95}$ using 6 additional characteristics. We again run our experiments for 2^{30} randomly selected pairs for 10000 keys and count the number of pairs following D. On average we would expect 530 pairs.

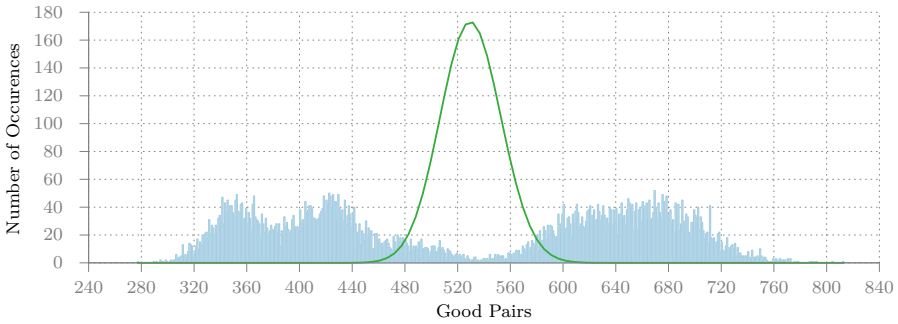

Fig. 7. Distribution of $\delta_K(D)$ over a random choice of K for 7-round `Speck`-64.

In Figure 7 it can be seen that for 7-round Speck-64 the distribution is bimodal and we over- respectively underestimate the number of valid pairs for most keys.

5.3 Midori

For Midori-64 we look at the differential

$$D = (\texttt{0x0200200000020000}, \texttt{0x0202220020020020})$$

over 4 rounds. The best characteristic in D has a probability of 2^{-32} and this improves to about $2^{-23.79}$ using 896 additional characteristics. We again run our experiments for 2^{30} randomly selected pairs for 3200 keys and count the number of pairs following D. On average we would expect about 74 pairs.

Fig. 8. Distribution of $\delta_K(D)$ over a random choice of K for 4-round Midori-64. We omitted the 2545 keys with 0 good pairs in this plot.

In Figure 8 it can be seen that for 4-round Midori-64 the distribution is very different from the previous cases. For some keys the probability is significantly higher and for about 80% of the keys we get 0 good pairs. This means that for a large fraction of keys we actually found an impossible differential and one should be careful when constructing differential distinguishers for Midori. In particular it would be interesting to classify this set of impossible keys and we leave this as an open problem. Moreover, this also implies the existance of a large class of weak keys, that has also been observed in the invariant subspace attacks on Midori-64 [34,27,49].

6 Conclusions

In this work we showed for several lighweight block ciphers that the gap between single characteristics and differentials can be surprisingly large. This leads to

significantly higher probability of differentials in several designs and allows us to have differential distinguishers covering more rounds.

We provided a simple framework to automate the process of collecting many differential characteristics that are contributing to the probability of a differential. We hope this will encourage future designs of cryptographic primitives to apply our methodology in order to provide better bounds on the security against differential cryptanalysis.

Further we verified differentials for a reduced number of rounds experimentally and showed that our improved estimates of the probability of differentials of `Skinny` closely resembles what happens in experiments. However, we can also observe that some commonly made assumptions on the distribution of good pairs following a differential over the choice of keys has to be made very carefully. For instance, the results for `Speck` and `Midori` indicate that one needs to be very careful in presuming that the estimates apply to all key values.

References

1. Abdelraheem, M.A., Ågren, M., Beelen, P., Leander, G.: On the distribution of linear biases: Three instructive examples. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology – CRYPTO 2012. Lecture Notes in Computer Science, vol. 7417, pp. 50–67. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2012)
2. Abed, F., List, E., Lucks, S., Wenzel, J.: Differential cryptanalysis of round-reduced Simon and Speck. In: Cid, C., Rechberger, C. (eds.) Fast Software Encryption – FSE 2014. Lecture Notes in Computer Science, vol. 8540, pp. 525–545. Springer, Heidelberg, Germany, London, UK (Mar 3–5, 2015)
3. Ankele, R., Banik, S., Chakraborti, A., List, E., Mendel, F., Sim, S.M., Wang, G.: Related-key impossible-differential attack on reduced-round skinny. In: Gollmann, D., Miyaji, A., Kikuchi, H. (eds.) ACNS 17: 15th International Conference on Applied Cryptography and Network Security. Lecture Notes in Computer Science, vol. 10355, pp. 208–228. Springer, Heidelberg, Germany, Kanazawa, Japan (Jul 10–12, 2017)
4. Ankele, R., List, E.: Differential cryptanalysis of round-reduced sparx-64/128. Cryptology ePrint Archive, Report 2018/332 (2018), `https://eprint.iacr.org/2018/332`
5. Aumasson, J.P., Jovanovic, P., Neves, S.: Analysis of NORX: Investigating differential and rotational properties. In: Aranha, D.F., Menezes, A. (eds.) Progress in Cryptology - LATINCRYPT 2014: 3rd International Conference on Cryptology and Information Security in Latin America. Lecture Notes in Computer Science, vol. 8895, pp. 306–324. Springer, Heidelberg, Germany, Florianópolis, Brazil (Sep 17–19, 2015)
6. Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., Regazzoni, F.: Midori: A block cipher for low energy. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology – ASIACRYPT 2015, Part II. Lecture Notes in Computer Science, vol. 9453, pp. 411–436. Springer, Heidelberg, Germany, Auckland, New Zealand (Nov 30 – Dec 3, 2015)
7. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK families of lightweight block ciphers. Cryptology ePrint Archive, Report 2013/404 (2013), `http://eprint.iacr.org/2013/404`

8. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology – CRYPTO 2016, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 123–153. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016)
9. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. In: Menezes, A.J., Vanstone, S.A. (eds.) Advances in Cryptology – CRYPTO'90. Lecture Notes in Computer Science, vol. 537, pp. 2–21. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 11–15, 1991)
10. Biryukov, A., Derbez, P., Perrin, L.: Differential analysis and meet-in-the-middle attack against round-reduced TWINE. In: Leander, G. (ed.) Fast Software Encryption – FSE 2015. Lecture Notes in Computer Science, vol. 9054, pp. 3–27. Springer, Heidelberg, Germany, Istanbul, Turkey (Mar 8–11, 2015)
11. Biryukov, A., Roy, A., Velichkov, V.: Differential analysis of block ciphers SIMON and SPECK. In: Cid, C., Rechberger, C. (eds.) Fast Software Encryption – FSE 2014. Lecture Notes in Computer Science, vol. 8540, pp. 546–570. Springer, Heidelberg, Germany, London, UK (Mar 3–5, 2015)
12. Biryukov, A., Velichkov, V.: Automatic search for differential trails in ARX ciphers. In: Benaloh, J. (ed.) Topics in Cryptology – CT-RSA 2014. Lecture Notes in Computer Science, vol. 8366, pp. 227–250. Springer, Heidelberg, Germany, San Francisco, CA, USA (Feb 25–28, 2014)
13. Blondeau, C., Nyberg, K.: Improved parameter estimates for correlation and capacity deviates in linear cryptanalysis. IACR Transactions on Symmetric Cryptology 2016(2), 162–191 (2016), http://tosc.iacr.org/index.php/ToSC/article/view/570
14. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2007. Lecture Notes in Computer Science, vol. 4727, pp. 450–466. Springer, Heidelberg, Germany, Vienna, Austria (Sep 10–13, 2007)
15. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knežević, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçin, T.: PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In: Wang, X., Sako, K. (eds.) Advances in Cryptology – ASIACRYPT 2012. Lecture Notes in Computer Science, vol. 7658, pp. 208–225. Springer, Heidelberg, Germany, Beijing, China (Dec 2–6, 2012)
16. Canteaut, A.: Differential cryptanalysis of feistel ciphers and differentially uniform mappings. Selected Areas on Cryptography, SAC'97 pp. 172–184 (1997)
17. Canteaut, A., Fuhr, T., Gilbert, H., Naya-Plasencia, M., Reinhard, J.R.: Multiple differential cryptanalysis of round-reduced PRINCE. In: Cid, C., Rechberger, C. (eds.) Fast Software Encryption – FSE 2014. Lecture Notes in Computer Science, vol. 8540, pp. 591–610. Springer, Heidelberg, Germany, London, UK (Mar 3–5, 2015)
18. Daemen, J., Lamberger, M., Pramstaller, N., Rijmen, V., Vercauteren, F.: Computational aspects of the expected differential probability of 4-round aes and aes-like ciphers. Computing 85(1), 85–104 (Jun 2009), https://doi.org/10.1007/s00607-009-0034-y
19. Daemen, J., Rijmen, V.: The wide trail design strategy. In: Honary, B. (ed.) 8th IMA International Conference on Cryptography and Coding. Lecture Notes in Computer Science, vol. 2260, pp. 222–238. Springer, Heidelberg, Germany, Cirencester, UK (Dec 17–19, 2001)

20. Daemen, J., Rijmen, V.: Plateau characteristics. IET Information Security 1(1), 11–17 (2007)
21. Davis, M., Logemann, G., Loveland, D.: A machine program for theorem-proving. Commun. ACM 5(7), 394–397 (Jul 1962), http://doi.acm.org/10.1145/368273.368557
22. Dinu, D., Perrin, L., Udovenko, A., Velichkov, V., Großschädl, J., Biryukov, A.: private communication
23. Dinu, D., Perrin, L., Udovenko, A., Velichkov, V., Großschädl, J., Biryukov, A.: Design strategies for ARX with provable bounds: Sparx and LAX. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology – ASIACRYPT 2016, Part I. Lecture Notes in Computer Science, vol. 10031, pp. 484–513. Springer, Heidelberg, Germany, Hanoi, Vietnam (Dec 4–8, 2016)
24. Dobraunig, C., Eichlseder, M., Kales, D., Mendel, F.: Practical key-recovery attack on MANTIS5. IACR Transactions on Symmetric Cryptology 2016(2), 248–260 (2016), http://tosc.iacr.org/index.php/ToSC/article/view/573
25. Eichlseder, M., Kales, D.: Clustering related-tweak characteristics: Application to mantis-6. IACR Transactions on Symmetric Cryptology 2018(2), 111–132 (2018), https://tosc.iacr.org/index.php/ToSC/article/view/890
26. Gérault, D., Lafourcade, P.: Related-key cryptanalysis of midori. In: Dunkelman, O., Sanadhya, S.K. (eds.) Progress in Cryptology - INDOCRYPT 2016: 17th International Conference in Cryptology in India. Lecture Notes in Computer Science, vol. 10095, pp. 287–304. Springer, Heidelberg, Germany, Kolkata, India (Dec 11–14, 2016)
27. Guo, J., Jean, J., Nikolic, I., Qiao, K., Sasaki, Y., Sim, S.M.: Invariant subspace attack against Midori64 and the resistance criteria for S-box designs. IACR Transactions on Symmetric Cryptology 2016(1), 33–56 (2016), http://tosc.iacr.org/index.php/ToSC/article/view/534
28. Jean, J., Nikolic, I., Peyrin, T.: Tweaks and keys for block ciphers: The TWEAKEY framework. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology – ASIACRYPT 2014, Part II. Lecture Notes in Computer Science, vol. 8874, pp. 274–288. Springer, Heidelberg, Germany, Kaoshiung, Taiwan, R.O.C. (Dec 7–11, 2014)
29. Joan Daemen, Michaël Peeters, Gilles Van Assche, Vincent Rijmen: Nessie Proposal: NOEKEON (2000), http://gro.noekeon.org/Noekeon-spec.pdf
30. Keliher, L., Sui, J.: Exact maximum expected differential and linear probability for two-round advanced encryption standard. IET Information Security 1(2), 53–57 (2007), https://doi.org/10.1049/iet-ifs:20060161
31. Kölbl, S., Leander, G., Tiessen, T.: Observations on the SIMON block cipher family. In: Gennaro, R., Robshaw, M.J.B. (eds.) Advances in Cryptology – CRYPTO 2015, Part I. Lecture Notes in Computer Science, vol. 9215, pp. 161–185. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 2015)
32. Kölbl, S., Roy, A.: A brief comparison of simon and simeck. In: Bogdanov, A. (ed.) Lightweight Cryptography for Security and Privacy. pp. 69–88. Springer International Publishing, Cham (2017)
33. Lai, X., Massey, J.L., Murphy, S.: Markov ciphers and differential cryptanalysis. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 547, pp. 17–38. Springer (1991)
34. Leander, G., Abdelraheem, M.A., AlKhzaimi, H., Zenner, E.: A cryptanalysis of PRINTcipher: The invariant subspace attack. In: Rogaway, P. (ed.) Advances in Cryptology – CRYPTO 2011. Lecture Notes in Computer Science, vol. 6841, pp. 206–221. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2011)

35. Leurent, G.: Analysis of differential attacks in ARX constructions. In: Wang, X., Sako, K. (eds.) Advances in Cryptology – ASIACRYPT 2012. Lecture Notes in Computer Science, vol. 7658, pp. 226–243. Springer, Heidelberg, Germany, Beijing, China (Dec 2–6, 2012)
36. Lipmaa, H., Moriai, S.: Efficient algorithms for computing differential properties of addition. In: Matsui, M. (ed.) Fast Software Encryption – FSE 2001. Lecture Notes in Computer Science, vol. 2355, pp. 336–350. Springer, Heidelberg, Germany, Yokohama, Japan (Apr 2–4, 2002)
37. Liu, G.Q., Jin, C.H.: Differential cryptanalysis of present-like cipher. Designs, Codes and Cryptography 76(3), 385–408 (Sep 2015), https://doi.org/10.1007/s10623-014-9965-1
38. Liu, G., Ghosh, M., Song, L.: Security analysis of SKINNY under related-tweakey settings (long paper). IACR Transactions on Symmetric Cryptology 2017(3), 37–72 (2017)
39. Liu, Z., Li, Y., Wang, M.: Optimal differential trails in SIMON-like ciphers. IACR Transactions on Symmetric Cryptology 2017(1), 358–379 (2017)
40. Mate Soos: CryptoMiniSat SAT solver (2009), https://github.com/msoos/cryptominisat/
41. Mouha, N., Preneel, B.: Towards finding optimal differential characteristics for ARX: Application to Salsa20. Cryptology ePrint Archive, Report 2013/328 (2013), http://eprint.iacr.org/2013/328
42. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Wu, C.K., Yung, M., Lin, D. (eds.) Information Security and Cryptology. pp. 57–76. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
43. Niemetz, A., Preiner, M., Biere, A.: Boolector 2.0 system description. Journal on Satisfiability, Boolean Modeling and Computation 9, 53–58 (2014 (published 2015))
44. Song, L., Huang, Z., Yang, Q.: Automatic differential analysis of ARX block ciphers with application to SPECK and LEA. In: Liu, J.K., Steinfeld, R. (eds.) ACISP 16: 21st Australasian Conference on Information Security and Privacy, Part II. Lecture Notes in Computer Science, vol. 9723, pp. 379–394. Springer, Heidelberg, Germany, Melbourne, VIC, Australia (Jul 4–6, 2016)
45. Stefan Kölbl: CryptoSMT: An easy to use tool for cryptanalysis of symmetric primitives (2015), https://github.com/kste/cryptosmt
46. Sun, S., Hu, L., Wang, M., Wang, P., Qiao, K., Ma, X., Shi, D., Song, L., Fu, K.: Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties. Cryptology ePrint Archive, Report 2014/747 (2014), http://eprint.iacr.org/2014/747
47. Suzaki, T., Minematsu, K., Morioka, S., Kobayashi, E.: TWINE : A lightweight block cipher for multiple platforms. In: Knudsen, L.R., Wu, H. (eds.) SAC 2012: 19th Annual International Workshop on Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 7707, pp. 339–354. Springer, Heidelberg, Germany, Windsor, Ontario, Canada (Aug 15–16, 2013)
48. Tezcan, C., Okan, G.O., Şenol, A., Doğan, E., Yücebaş, F., Baykal, N.: Differential attacks on lightweight block ciphers present, pride, and rectangle revisited. In: Bogdanov, A. (ed.) Lightweight Cryptography for Security and Privacy. pp. 18–32. Springer International Publishing, Cham (2017)
49. Todo, Y., Leander, G., Sasaki, Y.: Nonlinear invariant attack - practical attack on full SCREAM, iSCREAM, and Midori64. In: Cheon, J.H., Takagi, T. (eds.)

Advances in Cryptology – ASIACRYPT 2016, Part II. Lecture Notes in Computer Science, vol. 10032, pp. 3–33. Springer, Heidelberg, Germany, Hanoi, Vietnam (Dec 4–8, 2016)
50. Vijay Ganesh and Trevor Hansen and Mate Soos and Dan Liew and Ryan Govostes: STP constraint solver (2007), https://github.com/stp/stp
51. Wang, M., Sun, Y., Tischhauser, E., Preneel, B.: A model for structure attacks, with applications to PRESENT and Serpent. In: Canteaut, A. (ed.) Fast Software Encryption – FSE 2012. Lecture Notes in Computer Science, vol. 7549, pp. 49–68. Springer, Heidelberg, Germany, Washington, DC, USA (Mar 19–21, 2012)
52. Wang, N., Wang, X., Jia, K.: Improved impossible differential attack on reduced-round LBlock. In: Kwon, S., Yun, A. (eds.) ICISC 15: 18th International Conference on Information Security and Cryptology. Lecture Notes in Computer Science, vol. 9558, pp. 136–152. Springer, Heidelberg, Germany, Seoul, Korea (Nov 25–27, 2016)
53. Wang, X., Feng, D., Lai, X., Yu, H.: Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD. Cryptology ePrint Archive, Report 2004/199 (2004), http://eprint.iacr.org/2004/199
54. Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., Verbauwhede, I.: Rectangle: a bit-slice lightweight block cipher suitable for multiple platforms. Science China Information Sciences 58(12), 1–15 (Dec 2015), https://doi.org/10.1007/s11432-015-5459-7

A Detailed Data for Midori, Skinny and Sparx

In the following we give a more detailed overview over the analysis on Midori, Skinny and Sparx. In particular we give the following metrics

- Best differential characteristic for r rounds.
- Estimate of the differential with the input/output difference of the best differential characteristic found.
- Number of differential characteristics we used for the estimate.
- The maximum *weight* of the differential characteristics we use for the estimate.
- Search time to find the best single differential characteristic and all the differential characteristics for the best differential.

Table 3. Detailed results on the differentials found for `Midori`-64.

r	Pr_{char}	Pr_{diff}	#Characteristics	Max weight	$Time_{char}$	$Time_{diff}$
4	2^{-32}	$2^{-23.79}$	896	36	31m36s	2m4s
5	2^{-46}	$2^{-35.13}$	55168	54	56m42s	1h10m
6	2^{-60}	$2^{-48.36}$	11072	71	1h54m	29m
7	2^{-70}	$2^{-57.43}$	28588	99	3h12m	1h32m
8	2^{-76}	$2^{-60.87}$	693730	114	1h6m	23h36m
9	2^{-82}	$2^{-66.52}$	104694	90	56m	3h12m
10	2^{-100}	$2^{-83.86}$	120181	106	5h12m	4h36m
11	2^{-114}	$2^{-98.04}$	87055	119	10h56m	3h18m
12	2^{-124}	$2^{-108.59}$	88373	131	1d02h	4h54m
13	2^{-134}	$2^{-118.70}$	56596	139	22h02m	3h06m
14	2^{-144}	$2^{-131.18}$	13932	149	1d16h	9h36m
15	2^{-150}	$2^{-137.07}$	25680	155	20h30m	1h48m
16	2^{-168}	$2^{-155.58}$	11815	172	3d21h	1h12m

Table 4. Detailed results on the differentials found for `Skinny`-64.

r	Pr_{char}	Pr_{diff}	#Characteristics	Max weight	$Time_{char}$	$Time_{diff}$
6	2^{-32}	$2^{-23.51}$	100319	45	22m54s	1h38m
7	2^{-52}	$2^{-39.49}$	141800	58	1h03m	5h13m
8	2^{-72}	$2^{-56.55}$	821896	98	1h24m	23h20m
9	2^{-82}	$2^{-65.36}$	277464	89	1h06m	29h25m
10	2^{-92}	$2^{-75.98}$	66438	92	1h42m	2h59m
11	2^{-102}	$2^{-86.63}$	64339	103	2h36m	3h14m
12	2^{-110}	$2^{-95.00}$	62382	113	3h12m	3h37m
13	2^{-116}	$2^{-100.06}$	165079	124	2h42m	24h42m
14	2^{-122}	$2^{-106.71}$	100457	127	3h30m	10h25m
15	2^{-132}	$2^{-114.65}$	326404	142	7h23m	37h21m
16	2^{-150}	$2^{-135.41}$	24598	150	30h35m	1h44m
17	2^{-164}	$2^{-150.07}$	21524	165	60h09m	1h53m
18	2^{-176}	$2^{-161.64}$	20903	177	92h04m	1h54m
19	2^{-184}	$2^{-168.27}$	54245	185	60h22m	3h38m
20	2^{-192}	$2^{-176.74}$	39169	193	60h10m	2h59m
...						

B Differentials for `Midori`, `Skinny` and `Sparx`

In the following we give the best differentials that we found for `Midori`, `Skinny` and `Sparx`. The differentials for many other lightweight ciphers together with the source code to generate the differential models is publicly available at: `https://github.com/TheBananaMan/cryptosmt`

Table 5. Detailed results on the differentials found for `Sparx`-64.

r	$\Pr_{\text{char}}$	$\Pr_{\text{diff}}$	#Characteristics	Max weight	$\text{Time}_{\text{char}}$	$\text{Time}_{\text{diff}}$
1	1	1	1	1	0.02s	0.03s
2	2^{-1}	2^{-1}	1	2	0.1s	0.07s
3	2^{-3}	2^{-3}	1	4	0.5s	0.09s
4	2^{-5}	$2^{-4.99}$	8	49	2.4s	3.36s
5	2^{-9}	$2^{-8.99}$	12944	58	25s	2m12s
6	2^{-13}	$2^{-12.99}$	70133	51	3m48s	3h06m
7	2^{-24}	$2^{-23.95}$	56301	60	47h48m	28m
8	2^{-29}	$2^{-28.53}$	37124	60	15d5h	17m
9	2^{-35}	$2^{-32.87}$	233155	58	22d7h	7h42m
10	2^{-42}	$2^{-38.12}$	1294158	73	32d12h	35h18m
...						

Table 6. The best differentials that we found for various rounds of `Midori`-64.

r	Differential	$\Pr_{\text{Differential}}$
4	`0x0000020000022000` → `0x0020220002022002`	$2^{-23.79}$
5	`0x0004100000000100` → `0x0222220222222022`	$2^{-35.13}$
6	`0x0550000000005000` → `0x0000AA0000007707`	$2^{-48.36}$
7	`0x0AA00500700A0000` → `0x00005AFF0000AAA0`	$2^{-57.43}$
8	`0x0A000000A0000005` → `0x000000000000A0AA`	$2^{-60.87}$
9	`0x0000000A050000A0` → `0x770700000AAAA0AA`	$2^{-66.52}$
10	`0x0500005050000000` → `0xDD7A7D0D25727A7D`	$2^{-83.86}$
11	`0x0000A00000500500` → `0xAAA0AAA50AAAAA0A`	$2^{-98.04}$
12	`0xA0A00A0A00007000` → `0x0000DD7A00007077`	$2^{-108.59}$
13	`0x0000A0070A000AA0` → `0x00000555A5AFAF5F`	$2^{-118.70}$
14	`0x0000000000000500` → `0x000070777707AAA0`	$2^{-131.18}$
15	`0x0A0000A000000000A` → `0x05550000AA0AAAA0`	$2^{-137.07}$
16	`0xAA00A0A0AAA00A70` → `0x00007077AA0A7770`	$2^{-155.58}$

Table 7. The best differentials that we found for various rounds of `Skinny`-64.

r	Differential	$\Pr_{\text{Differential}}$
6	0x0041C00001000000 → 0x4044400400404444	$2^{-23.51}$
7	0x002220222B222000 → 0x0444004404004444	$2^{-39.49}$
8	0x0104401000C01C00 → 0x0606060000060666	$2^{-56.55}$
9	0x0020000200020200 → 0x0060000100600160	$2^{-65.36}$
10	0x0008200020000020 → 0x0008808000880088	$2^{-75.98}$
11	0x0002200000000200 → 0x0444004404004444	$2^{-86.63}$
12	0x0004000000000000 → 0x0001000100000001	$2^{-95.00}$
13	0x0200000000002000 → 0x0001001100000001	$2^{-100.06}$
14	0x4000040000400000 → 0x0404040000040444	$2^{-106.71}$
15	0x8008080000800000 → 0x1066100600601666	$2^{-114.65}$
16	0x0020000220000000 → 0x8880088080008888	$2^{-135.41}$
17	0x004C400004000000 → 0x2002022022020022	$2^{-150.07}$
18	0x400C0000C00C0000 → 0x0077001100660077	$2^{-161.64}$
19	0x2200000000002008 → 0x0077001100660077	$2^{-168.27}$
20	0x8800000000008009 → 0x8800080900008800	$2^{-176.74}$
...		

Table 8. The best differentials that we found for various rounds of `Sparx`-64.

r	Differential	$\Pr_{\text{Differential}}$
1	(0x0040, 0x8000, 0x0000, 0x0000) → (0x0000, 0x0002, 0x0000, 0x0000)	1
2	(0x0010, 0x2000, 0x0000, 0x0000) → (0x8000, 0x8002, 0x0000, 0x0000)	2^{-1}
3	(0x2800, 0x0010, 0x0000, 0x0000) → (0x8300, 0x8302, 0x8100, 0x8102)	2^{-3}
4	(0x0000, 0x0000, 0x2800, 0x0010) → (0x8000, 0x840A, 0x0000, 0x0000)	$2^{-4.99}$
5	(0x0000, 0x0000, 0x0211, 0x0A04) → (0x8000, 0x840A, 0x0000, 0x0000)	$2^{-8.99}$
6	(0x0000, 0x0000, 0x0211, 0x0A04) → (0xAF1A, 0xBF30, 0x850A, 0x9520)	$2^{-12.99}$
7	(0x0000, 0x0000, 0x7448, 0xB0F8) → (0x8004, 0x8C0E, 0x8000, 0x840A)	$2^{-23.95}$
8	(0x0000, 0x0000, 0x0050, 0x8402) → (0x0040, 0x0542, 0x0040, 0x0542)	$2^{-28.53}$
9	(0x2800, 0x0010, 0x2800, 0x0010) → (0x5761, 0x1764, 0x5221, 0x1224)	$2^{-32.87}$
10	(0x2800, 0x0010, 0x2800, 0x0010) → (0x8081, 0x8283, 0x8000, 0x8002)	$2^{-38.12}$
...		