State of the Art in Lightweight Symmetric Cryptography

Alex Biryukov[1] and Léo Perrin[2]

[1] SnT, CSC, University of Luxembourg, alex.biryukov@uni.lu
[2] Inria, Paris, leo.perrin@inria.fr

Abstract. Lightweight cryptography has been one of the "hot topics" in symmetric cryptography in the recent years. A huge number of lightweight algorithms have been published, standardized and/or used in commercial products.
In this paper, we discuss the different implementation constraints that a "lightweight" algorithm is usually designed to satisfy in both the software and the hardware case. We also present an extensive survey of all lightweight symmetric primitives we are aware of. It covers designs from the academic community, from government agencies and proprietary algorithms which were reverse-engineered or leaked. Relevant national (NIST...) and international (ISO/IEC...) standards are listed.
We identified several trends in the design of lightweight algorithms, such as the designers' preference for ARX-based and bitsliced-S-Box-based designs or simpler key schedules. We also discuss more general trade-offs facing the authors of such algorithms and suggest a clearer distinction between two subsets of lightweight cryptography. The first, *ultra-lightweight cryptography*, deals with primitives fulfilling a unique purpose while satisfying specific and narrow constraints. The second is *ubiquitous cryptography* and it encompasses more versatile algorithms both in terms of functionality and in terms of implementation trade-offs.

Keywords: Lightweight cryptography · Ultra-Lightweight · IoT · Internet of Things · SoK · Survey · Standards · Industry

1 Introduction

The Internet of Things (IoT) is one of the foremost buzzwords in computer science and information technology at the time of writing. It is a very broad term describing the fact that, in the near future, the Internet will be used more and more to connect devices to one another other rather than to connect people together.

Some of these devices use powerful processors and can be expected to use the same cryptographic algorithms as standard desktop PCs. However, many of them use extremely low power micro-controllers which can only afford to devote a small fraction of their computing power to security. Similarly, regular algorithms may incur too high a latency or too high a power consumption for such platforms.

A common example of such use is that of sensor networks. Such networks are intended to connect vast amounts of very simple sensors to a central hub. These sensors would run on batteries and/or generate their own energy using for example solar panels. Cryptographic algorithms must be used in the communication channels between the sensors and their hub in order to provide security, authenticity and integrity of the messages. However, because of the very low energy available, and because security is an overhead on top of the actual functionality of the device, the cryptographic algorithms have to be as "small" as possible. Similar reasoning goes for the size of ROM and RAM consumption of the cryptographic algorithm.

Similarly, RFID (Radio-Frequency IDentification) chips are used to identify devices, animals — and even people. In order to prevent an eavesdropper from learning the identification associated to a chip, this information has to be encrypted. Moreover, such RFID tags can be used in challenge response protocols. Because of the very small number of logical gates that can be used in such devices and because of the very little energy available, specially designed algorithms are necessary.

A vast number of symmetric cryptographic algorithms have been proposed to fill these use cases (see Table 1). They are usually referred to as "lightweight". Their designs vary greatly, the only unifying thread between them is the low computing power of the devices intended to run them.

Still, we call "lightweight" a wide range of algorithms with different properties and corresponding to very different use cases. As a consequence, it is difficult to give a clear definition of what this term entails. For example, does it make sense for a "lightweight" algorithm to be secure against attacks requiring a large amount of data? If the device using it is a simple RFID tag, probably not. Conversely, if it is an internet-enabled device downloading e.g. software updates at regular intervals, then such protection would be desirable.

Our Contributions. In this paper, we systematize the knowledge in the area of lightweight cryptography with the aim of better understanding what "lightweightness" is. Our work is based on an extensive literature survey of more than 100 algorithms[1], the exact number considered being summarized in Table 1. We list these algorithms and provide references for the best cryptanalysis against them. We also look at the broader eco-system of lightweight cryptography by discussing benchmarks as well as national and international standards. Finally, identify several trends in the design of lightweight algorithms, both in terms of choice of components and in terms of high level scope statement. In particular, we argue that *lightweight cryptography* could be split into *ultra-lightweight cryptography* and *ubiquitous cryptography*. The former deals with highly specialized algorithms providing one function with high performance on one platform; for instance stream ciphers optimized for low area implementations on ASICs. On the other hand, *ubiquitous cryptography* is about more versatile algorithms, both in terms of functionality and implementation; for instance sponge permutation that can be implemented fairly efficiently on ASICs, FPGAs, micro-controllers, etc.

Table 1: The number of lightweight symmetric algorithms surveyed in this paper. Note that some authenticated cipher were published along with a new a block cipher; those appear in the "Auth. C." column.

	Stream C.	Block C.	Hash F.	Auth. C.	MAC	Total
Academia †	15	48	10	14	2	89
Proprietary	16	5	0	0	1	22
Government	1	5	0	0	0	6
Total	32	58	10	14	3	**117**

† Here and in the rest of the paper we count in "academia" all cryptographic primitives which were published in peer-reviewed academic articles together with clear design rationale, including those from the industry.

[1]Since the area of LW cryptography is very large and still actively growing we welcome the reader to point us to new (and old) results that we could have missed, to make this survey more complete.

Outline. Section 2 lists the design constraints that make an algorithm lightweight. Both the hardware and software cases are described. We also discuss the benchmark of these algorithms. Then, we list all symmetric cryptographic lightweight algorithms we are aware of in Section 3. We consider algorithms published at cryptography/security conferences as well as those designed by government agencies, those specified in standards and those designed by the industry which were reverse-engineered from actual products. In particular, the following tables list lightweight algorithms along with a reference for their specification and some of their properties such as key and block sizes:

- Table 3 lists proprietary ciphers. None of them provides more than 64 bits of security either by design (short key), because of weaknesses in their design, or both;
- algorithms published at cryptography/security conferences or submitted to cryptography competitions are listed in Table 4 (stream ciphers), Table 6 (block ciphers), Table 5 (hash functions), Table 7 (authenticated ciphers), Table 8 (MACs); and
- Table 9 lists public ciphers designed by government agencies.

We also survey the standardization of lightweight crypto by both national and international bodies in Section 4.

In Section 5, we investigate some trends in the field of lightweight cryptography such as the increasing popularity of lighter key schedules or of the sponge structure. In fact, we list all sponges intended for lightweight cryptography in Table 11. Finally, we discuss some of the trade-offs at the core of lightweight cryptography in Section 6 and suggest a division of this huge area of research. Section 7 concludes our paper.

2 Design Constraints

The metrics usually optimized are the memory consumption, the implementation size and the speed or throughput of the primitive. However, the specifics of the comparison depend on whether hardware or software implementations are considered. More generally and regardless of the platform, an implementation is a trade-off between security, performances and cost. A cheap and efficient chip may be vulnerable to side-channel attacks; a cheap side-channel-resilient one may be slow and a fast and secure one can be expected to be expensive. The role of lightweight algorithms is to provide better trade-offs in this space by e.g. lowering the difficulty of a very efficient hardware implementation.

2.1 Hardware Case

If the primitive is implemented in hardware, the following metrics describe the efficiency of the implementation.

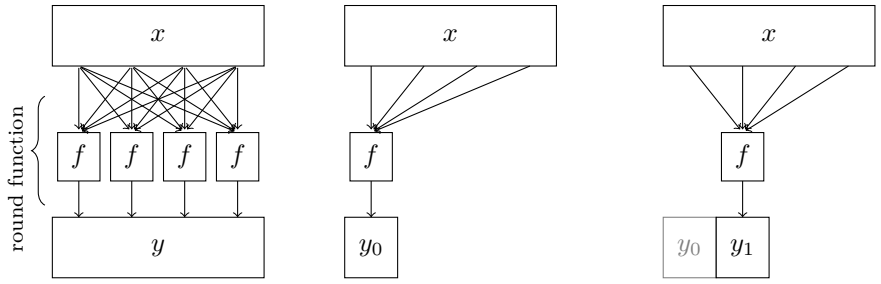
- The memory consumption and the implementation size are lumped together into its *gate area* which is measured in Gate Equivalents (GE). It quantifies how large the physical area needed for a circuit implementing the primitive is. The lower it is, the better.
- The throughput, measured in bits or bytes per second, corresponds to the amount of plaintext processed per time unit. The higher it is, the better.
- The latency, measured in seconds, correspond to the time taken to obtain the output of the circuit once its input has been set. The lower it is, the better.
- The power consumption, measured in Watts, quantifies the amount of power needed to use the circuit. The lower it is, the better.

These four criteria compete with one another. For instance, a low latency tends to imply a higher area. Different implementations of the AES are compared in [BDE+13] and the by far smallest implementation is also by far the slowest while the most energy efficient are the largest. The optimal trade-off between these quantities is very much context dependent. In fact, primitives have been proposed that have been optimized for different corners of the design space: some allow a very low latency implementation, others a very small one (in terms of GE), etc.

Regardless of the exact platform, a given primitive may be implemented using different approach. Let us consider round-based permutations—which may be block ciphers, sponge permutations, the inner component of a hash function, etc. A direct approach consists in storing the full internal state (along with the key state, if any) and then perform one round using a circuit operating on the full state at once. This is called a *round-based implementation*. In this case, each output bit is evaluated in parallel as summarized in Figure 1a. It is also possible to push this logic further and, rather than doing rounds one after another, to do several at once instead. Such unrolled implementations are popular when a low-latency is targeted as those allow a full evaluation in one clock cycle. The downside is then the far larger size of the circuit.

Serial implementations work differently. They only update a small part of the state at a time—typically the size of an S-Box output—as summarized in Figures 1b and 1c. At $t = 0$, only the input register x is set. At $t = 1$, after the first clock cycle, the whole state of the parallel implementation is set but only a part of the output of the serial implementation is. At $t = 2$, another part of the output of the serial implementation is set. In the example in Figures 1b and 1c, the full output will be set after $t = 4$ clock cycles.

Due to the simplicity of the circuit logic involved, serialized implementations allow a far higher clock frequency so they may not be as slow as one might fear. Still, their main advantage is a much smaller circuit.



(a) Round-based ($t = 1$). (b) Serial ($t = 1$). (c) Serial ($t = 2$).

Figure 1: Some implementation strategies in hardware.

Memory is often the most expensive part of the implementation of a lightweight primitive. As a consequence, it is preferable to operate on small blocks using a small key. However, some space can be saved by hard-coding or "burning" the keys into the device. That is, instead of using read/write memory to store the key, use read-only structures. In order for this method to be viable, the key schedule must build the round keys using only simple operations taking as input the bits of the master key. In particular, no key state can be operated upon. This strategy has been used by several algorithms, both block and stream ciphers, as shown in Section 5.3.2.

Lightweight algorithms have been proposed to fill different niches of the design space. For example, energy and power efficiency of the hardware implementation are at the core of the design of the Midori block cipher [BBI+15]. A very low latency demands specific design choices as illustrated by the lightweight block ciphers PRINCE [BCG+12] followed by Mantis [BJK+16] and Qarma [Ava17].

Platform Examples. One of the most common example were a hardware implementation is needed is the case of RFID tags. These correspond to small circuits which are very cheap and sometimes passively powered. Due to their very low cost, it makes sense to try and reduce the implementation cost of the cryptographic primitive used as much as possible. Besides, due to the low power they have available, it is necessary to use an algorithm which consumes as little power as possible.

The hardware efficiency is also relevant if (part of) the algorithm is to be provided by a dedicated instruction on a processor. For example, recent computer processors provide instructions performing AES rounds in one clock cycle. This *hardware acceleration* is also provided by some smaller processors.

2.2 Software Case

Lightweight primitives can be also implemented in software, typically for use on micro-controllers. In this case, the relevant metrics are the RAM consumption, the code size and the throughput:

- RAM consumption corresponds to the amount of data which is written to memory during each evaluation of the function,
- code size is the fixed amount of data which is needed to evaluate the function independently from its input, and
- as before, the throughput measures the average quantity of data which is processed during each clock cycle.

The first two are measured in bytes and should be minimized while the latter is measured in bytes per cycle and should be maximized. Again, these three quantities are not independent. For example, loading information from the RAM into CPU registers is a costly operation and so is its inverse. Therefore, limiting the number of such operations leads to a decrease in both RAM consumption *and* an increase in throughput.

The fact that micro-controllers operate on small words of 8, 16 or 32 bits means that some operations can have counter intuitive costs. On most 32-bit processors, all 32-bit rotations have the same (low) cost. However, 8-bit micro-controllers do not share this property. Rotations by multiples of 8 are quite cheap as they merely correspond to shuffling 8-bit registers. On the other hand, there is no dedicated instruction to perform other rotations 32-bit rotations. As a consequence, rotations have very different and non-negligible prices. This stands in stark contrast from the "non-lightweight" setting, that is, algorithms running on desktop computers and internet servers. It is also quite different from the hardware case where all bit permutations are extremely cheap.

Much like in the hardware case, there are different trade-offs and implementation techniques. For example, the subkeys of a block cipher can be precomputed to save time at the cost of memory during an encryption. It is also possible to use much more sophisticated techniques like bit-slicing which, while they may not always be applicable, can lead to substantial performance gains as shown in [SS16]. In fact, some recent lightweight algorithms have been explicitly designed to allow an efficient bit-sliced implementation on regular desktop computers such as SKINNY [BJK+16] and GIFT [BPP+17].

2.3 Side-Channel Attack Resilience

Side-channel attacks (SCAs) use some special knowledge about the implementation of a cipher to break its security. For example, observing the power consumption of an encryption can leak information about the Hamming weight of the output of an S-Box.

Such attacks demand that the cryptanalyst has physical access to the device attacked. This requirement is particularly easy to fulfill in the context of the IoT: a desktop computer

can be expected to be reasonably hard to physically interact with because it is in a locked room but a sensor measuring traffic in an open street may not enjoy such protection. As a consequence, lightweight algorithms are often built in such a way as to decrease the vulnerability of their implementation to such attacks.

Very generally, SCAs exploit the fact that the different operations performed during an encryption correspond to physical processes which can leak information about a secret such as the key. For example, if S is a small 8-bit S-Box, then the power consumption of a given implementation during the evaluation of $S(0)$ may be different from the one during the evaluation of $S(128)$. As a consequence, measuring precisely the power consumption of this device may reveal some information about the input of the S-Box and, thus, about the internal state of the cipher. Power consumption is but one of the many quantities that can leak such information; electromagnetic radiations, precise timing and even sound have been successfully used as well. Both hardware and software implementation can be vulnerable.

Even though the exact form of the leakage depends on the platform used to implement the cipher, the high level design of the primitive itself can simplify a less leaky implementation. This can be done through the use of inherently less leaky operations or by reducing the cost of a masked implementation.

Different operations leak different amounts of information. For example, a table look-up over 8-bit leaks more data than a simple AND. Thus, it may be preferable to use the latter rather than the former to prevent SCAs.

However, all operations can be *masked*. This implementation technique uses an external source of random data to randomize the input of the different operations that might leak information. More formally, it ensures that the input of each operation is statistically independent from secret data such as the internal state of a block cipher. For example, in a seminal paper on this topic [GP99], Goubin and Patarin proposed to modify the evaluation of the DES as follows. The 64-bit input block x is replaced by two blocks x' and x'' such that $x = x' \oplus x''$. This property is preserved by the bit permutations used by this cipher. Each of the eight different 6-to-4 bits S-Boxes S_i is replaced by a pair of 12-to-4 bits functions A_i, f_i, where $f_i(v', v'') = S_i(v' \oplus v'') \oplus A_i(v', v'')$. When an S-Box input v is split into $v = v' \oplus v''$, such functions verify $f_i(v', v'') \oplus A_i(v', v'') = S_i(v)$, meaning that they can be used to properly evaluate S_i while only evaluating functions with randomized inputs.

Different operations can be more or less easy to mask. Thus, it is possible and indeed becomes a trend in lightweight cryptography to simplify the implementation of such counter-measures by designing a cipher using exclusively such operations. For example, these considerations play a crucial role in the choice of the non-linear layer, as discussed in Section 5.1.

2.4 On Benchmarking

In order to assess whether a given algorithm has good performances on a given platform or, similarly, to find out what algorithm is the best in a given context, it is necessary to have a unique benchmarking tool to provide a unified and fair comparison. Designing such a program is a subtle and difficult task as requires to take into account a collection of edge cases. For example, if the block cipher uses an "on-the-fly" key schedule (see Section 5.3.3) then the key state must be modifiable during encryption and the RAM consumption of this added functionality must be measured. It is also necessary to consider multiple platforms which will each require a hand-optimized implementation of each algorithm.

2.4.1 Software Benchmarking

Several frameworks are dedicated to benchmarking the software implementation of cryptographic primitives. SUPERCOP[2] provides thorough benchmarks for different kinds of primitives on a wide variety of computers but, as is, it does not study micro-controller implementations. It was thus extended by the "eXternal Benchmarking eXtension" (XBX) [WBG10] to add support for several micro-controllers. The corresponding `github` page[3] contains detailed results about the finalists of the SHA-3 competition.[4]

Other frameworks are dedicated specifically to lightweight software implementation but also narrow their scope down to block ciphers only. For example, the benchmark of the BLOC project [CMM13] looked at the performances of many lightweight block ciphers on a 16-bit micro-controller (the MSP430 of Texas Instrument). Similarly, the work presented in the latest CRYPTREC report on lightweight cryptography compares implementations of several lightweight block ciphers [CRY17, Section 3.1.2.2] and authenticated ciphers [CRY17, Section 3.2.1.1] on the same 16-bit processors (RL78 of Renesas Electronic).

The FELICS framework [DBG+15] ("Fair Evaluation of LIghtweight Cryptographic Systems") allows a comparison of the RAM consumption, code size and throughput across algorithms, across different implementations of a given algorithm and across three different platforms. FELICS takes as input the implementation of a block or stream cipher and outputs the corresponding code size, RAM consumption and time taken to perform a given task. These quantities are obtained for three different micro-controllers: an 8-bit AVR, a 16-bit MSP and a 32-bit ARM. The tasks investigated correspond to different scenarios such as the encryption of a 128-bit block in counter-mode. The information extracted is then summarized into a single quantity called Figure of Merit (FoM), the lower the better. It can be used to rank block ciphers, as shown in Table 2 where the block and key sizes are in bits, the code size and maximum RAM consumption are in bytes and the time is in number of CPU cycles.

Table 2: The current best FELICS results for scenario 2: counter mode encryption of 128 bits.

General info			AVR (8-bit)			MSP (16-bit)			ARM (32-bit)			FoM
Name	block	key	Code	RAM	Time	Code	RAM	Time	Code	RAM	Time	
Chaskey	128	128	770	84	1597	490	86	1351	178	80	614	4.7
SIMON	64	96	448	53	2829	328	48	1959	256	56	1003	4.8
SIMON	64	128	452	53	2917	332	48	2013	276	60	972	4.9
Chaskey-LTS	128	128	770	84	2413	492	86	2064	178	80	790	5.4
SIMON	64	96	600	57	4269	460	56	2905	416	64	1335	6.6
SIMON	64	128	608	57	4445	468	56	3015	388	64	1453	6.8
Lea	128	128	906	80	4023	722	78	2814	520	112	1171	7.6
Rectangle	64	128	602	56	4381	480	54	2651	452	76	2432	8.1
Rectangle	64	80	606	56	4433	480	54	2651	452	76	2432	8.1
SPARX	64	128	662	51	4397	580	52	2261	654	72	2338	8.3
SPARX	128	128	1184	74	5478	1036	72	3057	1468	104	2935	12.4
RC5-20	64	128	1068	63	8812	532	60	15925	372	64	1919	13.5
AES	128	128	1246	81	3408	1170	80	4497	1348	124	4044	14.1
Hight	64	128	636	56	6231	636	52	7117	670	100	5532	14.8
Fantomas	128	128	1712	76	9689	1920	78	3602	2184	184	4550	18.8
Robin	128	128	2530	108	7813	1942	80	4913	2188	184	6250	22.0

[2]The webpage of this project is https://bench.cr.yp.to/supercop.html.
[3]https://github.com/das-labor/xbx
[4]https://github.com/das-labor/xbx/blob/master/documentation/benchmarking_results_may_2012.pdf

2.4.2 Hardware Benchmarking

A fair comparison of hardware implementation is even harder than a software one. Indeed, the exact values of the different performance metrics depend on the exact technology considered. Furthermore, the tools used to simulate those circuits do not give the same results and are often both proprietary and expensive. Therefore, a fair comparison of the different algorithms proposed regarding their hardware implementation is very difficult. In fact, when comparing their new algorithm with existing ones, designers are often forced to design their own implementations of preexisting ones too.

Nevertheless, significant efforts have been put towards a fair comparison of the hardware implementation of some algorithms, for example the authenticated ciphers submitted to the CAESAR competition. The ATHENa project [GKA+10] of the George Mason University provides detailed figures for the FPGA implementation of all CAESAR candidates[5] including those explicitly aiming at lightweightness. These algorithms are among the authenticated ciphers listed in Section 3.2.4. The exact ranking of the algorithms depends on the platform used but ACORN, Ascon, some variant of Ketje, and NORX are usually the best in terms of throughput/area on all platforms considered by ATHENa.

The CRYPTREC report on lightweight cryptography [CRY17, Section 3.1.1.1] also contains very thorough benchmarks of multiple implementations (unrolled, round-based and serial) of several block ciphers. Circuits considering encryption only and both encryption and decryption are considered. Multiple metrics are extracted: area, throughput, peak power consumption, etc. Unlike in the software case, it is difficult to summarize the properties of an algorithm using a single figure of merit. We thus refer the readers to [CRY17, Section 3.1.1.1] for more details.

2.5 Are Dedicated Algorithms Needed?

As lightweightness is mostly a property of the implementation of an algorithm, we can wonder if dedicated algorithms are actually needed. Would it not be sufficient to use lightweight *implementations* of regular algorithms?

It is often possible. For example, many implementers have worked on optimizing the implementation of the AES with some success in both hardware [BJM+14, BBR16a, UMHA16] and software [OBSC10, MM15, SS16]. Furthermore, some devices are sometimes shipped with a hardware acceleration module providing AES encryption and decryption, effectively adding a new set of instructions dedicated entirely to a quick evaluation of this block cipher.

In this context, lightweight symmetric algorithms may seem unnecessary. However, block cipher hardware acceleration has its limitations. As summarized for example in Table 1 of [OC16], the hardware-accelerated encryption used by many devices are vulnerable to various forms of side-channel attacks. These attacks do not only target these devices "in a vacuum". For example, Philips light bulbs using the Zigbee protocol to communicate have been shown to be insecure [ROSW16]. One of the key components of this attack is a subversion of the update mechanism of the light bulb. Updates are normally authenticated with an AES-based MAC using a secret key which is constant across all devices. Ronen *et al.* recovered this key via an SCA and were therefore able to push malicious updates to these devices.

On micro-controllers without hardware support for cryptographic functions, the implementation of the primitives used must be as small and as fast as possible. In these cases, the AES is rather fast; in fact, it is one of the fastest on 8-bit micro-controllers. However, its implementation requires storing at least the full look-up-table of its 8-bit S-Box or the

[5] Very detailed results can be found on the project's dedicated webpage: https://cryptography.gmu.edu/athenadb/fpga_auth_cipher/rankings_view.

full code corresponding to its bit-slice implementation, neither of which can be made very small.

All in all, while the AES is a reasonably lightweight block cipher, its large S-Box, large block size and inherent vulnerability to SCA caused by its look-up-based S-Box mean that there are extreme design goals that it is unlikely to fulfill such as a GE count under 2000, a very small code size or a very efficient masked implementation. This does not mean that the AES should not be used or that any lightweight cipher is an improvement—again, it is one of the fastest on 8-bit processors—but rather that there are algorithms capable of offering significant performance improvements in a cheap way.

Another case where dedicated lightweight algorithms are needed is for hashing. Indeed, standard hash functions need large amounts of memory to store both their internal states—1600 bits in the case of SHA-3—and the block they are operating on—512 bits in the case of the SHA-2 family. These memory requirements significantly hinder performance on lightweight platforms and justify the need for dedicated lightweight hash functions.

3 A Survey of Symmetric Lightweight Algorithms

Several very distinct actors are involved in the field of lightweight cryptography. In fact, they are the same that discuss "regular" cryptography: academia, industry, standardizing bodies and government agencies — including spying agencies.

The industry implements such algorithms in its products, designing or choosing the best one for each purpose. Unfortunately, until the 2000's and the spread of the AES, many of the algorithms implemented in practice had been designed in-house, with little regard to what is now considered best practice. The corresponding algorithms are described in Section 3.1.

Dozens of symmetric cryptographic algorithms claiming to be lightweight have been published at academics conferences. Those are listed in Section 3.2. We also list the best attacks against these algorithms, several of which are vulnerable to full-round cryptanalysis.

Academic publications always contain a description of the cryptanalysis attempts by the authors of the algorithm. This creates a significant contrast with the algorithms proposed by government agencies: even their public algorithms have been designed in a secret way. Furthermore, these agencies are more often than not also in charge of spying (e.g. the American NSA and the Russian FSB) meaning that they have contradictory incentives. This dichotomy is best illustrated by the interventions of the NSA in the cryptographic standardization of two algorithms. First in the standardization of IBM's design as the Data Encryption Standard (DES) in the 70's. On the one hand, they imposed a reduction of the key size of the future cipher from 128 to 56 bits while, on the other hand, supervising modifications of the design of the DES to ensure its resilience against the differential attack which was at the time unknown in academia. The other example is the role played by the NSA in the standardization by the NIST of the easily trapdoored Dual EC pseudo-random number generator [BLN15]. This subversion of a public standard backfired when an attacker placed a backdoor in the PRNG of a product of Juniper Networks, as detailed in [CMG+16].

We list lightweight algorithms designed by government agencies in Section 3.3.

3.1 Proprietary/Legacy Algorithms

Many lightweight algorithms used by industrial products are surprisingly weak. Many of those algorithms were designed in the 80's or early 90's, a time during which the public knowledge on cryptography was a mere fraction of what it is today. Cipher design also had to accommodate for the stringent American export laws which forbid selling devices

with strong cryptography. Still, like modern lightweight algorithms, those were intended to run on devices with little computing power devoted to encryption.

Table 3: A summary of some proprietary/legacy lightweight primitives. Block ciphers are marked with "†", MACs with "‡" and unmarked primitives are stream ciphers.

Name	Intended platform	Key	IS	IV	Att. time	Reference
A5/1	Cell phones	64	64	22	2^{24}	[And94]
A5/2		64	81	22	2^{16}	[BBK08]
CMEA †		64	16–48	–	2^{32}	[WSK97]
ORYX		96	96	–	2^{16}	[WSD+99]
A5-GMR-1	Satellite phones	64	82	19	$2^{38.1}$	[DHW+12]
A5-GMR-2		64	68	22	2^{28}	[DHW+12]
DSC	Cordless phones	64	80	35	2^{34}	[LST+09]
SecureMem.	Atmel chips	64	109	128	$2^{29.8}$	[GvRVWS10]
CryptoMem.		64	117	128	2^{50}	
Hitag2	Car key/ immobilizer	48	48	64	2^{35}	[VGB12]
Megamos		96	57	56	2^{48}	[VGE13]
Keeloq †		64	32	–	$2^{44.5}$	[BSK96]
DST40 †		40	40	–	2^{40}	[BGS+05]
iClass	Smart cards	64	40	–	2^{40}	[GdKGV14]
Crypto-1		48	48	96	2^{32}	[NESP08]
CSS	DVD players	40	42	–	2^{40}	[BD04]
Cryptomeria †		56	64	–	2^{48}	[BKLM09]
CSA-BC †	Digital televisions	64	64	–	2^{64}	[WW05]
CSA-SC		64	103	64	$2^{45.7}$	
PC-1	Amazon Kindle	128	152	–	2^{31}	[BLR13]
SecurID ‡	Secure token	64	64	–	2^{44}	[BLP04]
E0	Bluetooth devices	128	128	–	2^{27}	[ZXF13]

They were also intended to be kept secret, their secrecy hopefully enhancing their security. However, they were eventually released through leaks or reverse-engineering.[6] In a clear vindication of Kerchkoffs' law [Ker83], they were broken as soon as they were made public. Attacks with a time complexity under 2^{40} evaluations of the primitive exist for all of these algorithms except for CSA ciphers. Many of them are the target of even more powerful cryptanalyses. It should be noted that many of these algorithms are fairly old, meaning that they could not possibly be designed to resist forms of cryptanalysis that were not known at the time of their conception. Still, this highlights another problem: once deployed, algorithms are used for a long time which may very well extend beyond the date they are broken. This shows the difficulty of balancing the need for performances with the need for a security margin which can be hoped to last for a sufficient time; a conundrum which is at the core of the design of cryptographic primitives and even more so for lightweight algorithms.

The proprietary primitives are listed in Table 3 and short descriptions are provided in Appendix A. Block ciphers are marked with "†", MACs with "‡" and unmarked primitives are stream ciphers. The internal state (IS), key and initialization vector (IV) sizes are expressed in bits. For block ciphers, the internal state size corresponds to the block size. The time complexity of the best attack targeting the full round primitive is given in the column "Att. time". Our aim with this figure is to illustrate the efficiency of these attacks, a reader interested in the details of their efficiency (in particular regarding their data

[6] Except for E0 and for PC-1.

complexity) is encouraged to read the references provided.

It should be noted that we listed proprietary algorithms. There are algorithms from the industry which have been published at cryptography or security conferences such as TWINE or Enocoro. These are listed in the appropriate sections below.

3.2 A Semi-Exhaustive List of Public Algorithms

Throughout the last 25 years and especially since 2011, a lot of algorithms intended to be lightweight have been published in cryptography- and security-related conference proceedings and journals. Those are listed in this section.

The algorithms in these lists have either been advertised as lightweight in their specification, have a very small implementation or have been standardized as such. Figure 2 provides an overview of all lightweight symmetric algorithms published in the literature sorted by publication date and by type. They have usually been designed by academics but some companies also publish their algorithms in the same way. The categories depend on the type of algorithm published but, of course, the block cipher used in a block cipher-based authenticated cipher could be used on its own. Similarly, the permutation used in a sponge-based hash function could be used to build an authenticated cipher.

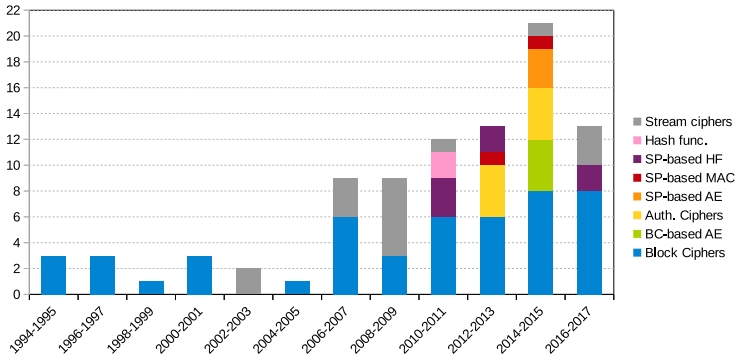


Figure 2: Lightweight algorithms published by academics. Block cipher-based authenticated ciphers are listed as "BC-based AE"; sponge-based algorithms are shown as such.

3.2.1 Stream Ciphers

The eSTREAM competition was held in 2008 to choose two portfolios of stream ciphers. The first type of algorithms fits the so-called *software profile*, meaning that they were aimed at software efficiency. The second category was the *hardware profile*. Further, some of them use an internal state so small that they can be considered to be lightweight stream ciphers.

However, lightweight stream ciphers were proposed outside the framework of this competition. For example, SNOW 3G corresponds to a simple modification of the academic-designed SNOW 2.0 tailored for specific industrial needs: it is used in the 3GPP communication standard. All such lightweight stream ciphers we are aware of are listed in Table 4.

We marked the stream ciphers vulnerable to attacks against the full primitive with a "†" symbol. A2U2 and Sprout are vulnerable to guess-and-determine attack presented respectively in [ABZD11] and in [CLNP16]. Earlier versions of the F-FCSR-H and F-FCSR-16 stream ciphers, which were selected for the eSTREAM portfolio, were broken in practical time by Hell and Johansson [HJ11]. Similarly, "version 0" of Grain was broken by Berbain *et al.* [BGM06]. Still, Grain-128 has a significant weak key class of density

Table 4: A summary of all lightweight stream ciphers from academia we are aware of, sorted by publication date. The key, internal state (IS) and initialization vector (IV) sizes are expressed in bits.

Name	Reference	Key	IV	IS
BeepBeep	[Dri02]	192/224	32	224
SNOW 2.0	[EJ03]	128/256	128	576
SNOW 3G	[ETS06]	128	128	576
Trivium	[Can06]	80	80	288
Grain †	[HJM07]	80/128	64/96	160/256
Chacha20	[Ber08a]	256	64	256
Enocoro-80	[WIK+08]	80	64	176
MICKEY v2	[BD08]	80/128	0–80/0–128	200/320
Rabbit	[BVCZ08]	128	64	513
Salsa20	[Ber08b]	512	64	512
F-FCSR-H/16 v3	[ABL+09]	80/128	80/128	160/256
A2U2 †	[DRL11]	61	64	95
Sprout † ‡	[AM15]	80	70	89
Espresso	[DH17]	128	96	256
LIZARD †	[HKM17]	120	64	121
Plantlet ‡	[MAM17]	80	90	110

†While LIZARD uses a 120-bit key, its designers only claim 80-bit security.
‡The key is stored separately from the internal state in non-volatile memory.

2^{-10} which can be attacked using dynamic cubes [DS11]. Attacks against reduced variants of some other stream ciphers exist but, since the encryption does not consist in iterating a round several times, we cannot define the fraction of the cipher that is broken as we do below for block ciphers.

3.2.2 Hash functions

It is more difficult to implement a lightweight hash function than a lightweight block cipher. Indeed, they usually require a much larger internal state which is reasonable on a desktop computer but would have a prohibitive cost on a lightweight device. For example, SHA-3 uses a 1600-bit internal state which dwarfs the 64-bit block of most lightweight block ciphers. Lightweight primitives based on Keccak exist such as Ketje (see Section 3.2.4) and their main difference with SHA-3 is indeed a significant reduction of the internal state size, down from 1600 bits to e.g. 200 bits. This modification is described—though it is not part of SHA-3—in [U.S15].

All lightweight hash functions we are aware of are in Table 5. The internal state (IS) size refers to the size of the chaining value. Sponge-based hash functions are marked with a "‡" symbol. It should be noted that the latest two sponge-based hash functions, namely GIMLI-hash and the hash function using sLiSCP, were introduced as multi-purpose permutations intended to be used to build sponge-based hash functions as well as sponge-based authenticated encryption—or any other sponge-based primitive.

Overall, lightweight hash functions have received less attention from cryptanalysts than lightweight block cipher. Nevertheless, attacks have been found against Armadillo2 (one of the two variants of Armadillo) [ABN+11] and the smallest instance of GLUON [PK15]. We thus marked these primitives with a "†" mark. There is a distinguisher for the full internal block cipher of Lesamnta-LW [SA12] but its consequences for the hash function itself are unclear.

Table 5: A summary of all lightweight hash functions from academia, sorted by publication date. The digest, internal state (IS) and block sizes are expressed in bits.

Name	Reference	Digest	Block/Rate	IS
Armadillo †	[BDN+10]	80/128/160/256	48/64/80/128	256/384/576/768
QUARK ‡	[AHMN10]	136/176/256	8/16/32	136/176/256
Lesamnta-LW	[HIK+11]	256	128	256
PHOTON ‡	[GPP11]	80/128/160/224/256	16/32/36	100/144/196/256/288
Spongent ‡	[BKL+11]	80/128/160/224/256	8/16	88/136/176/240/272
GLUON ‡ †	[BDM+12]	128/160/224	8/16/32	136/176/256
SPN-Hash ‡	[CYK+12]	128/256	256/512	128/256
GIMLI-hash ‡	[BKL+17]	256	128	384
sLiSCP ‡	[ARH+17]	160/192	32/64	192/256

3.2.3 Block Ciphers

Block ciphers are the most common choice for academic designers trying to build a lightweight symmetric algorithm. All those designed and published by academics we are aware of are listed in Table 6 where they are sorted by date of publication. We use "†" to mark block ciphers published as part of a CAESAR[7] submission and "‡" to indicate tweakable block ciphers.

We also list the best attacks against each block cipher in Table 6. For each cipher, we looked for the attack breaking the largest fraction of the rounds across all versions of the cipher and wrote down this fraction for single-key attacks (SK) and related-key attacks (RK) in two different columns. We did not considered attacks where the main loop goes through all possible keys like biclique attacks. For tweakable block ciphers, we consider related-tweak attacks as single-key attacks if the tweak and key are separated and as related key attacks when the TWEAKEY framework was used to build the cipher.

We use the $\emptyset$ symbol when no attack has been described. It does *not* imply that the author did not analyze their primitive, only that neither the authors nor anyone else has provided a clear attack against a (round-reduced) version of the algorithm. For example, if it is shown that r rounds are sufficient to prevent the existence of differential attacks and if the block cipher uses $2r$ rounds, it may be unnecessary for the designers to actually write down a complete differential cryptanalysis.

The existence of (related-key) attack against the full-round primitive is not sufficient to discard its usage. Indeed, while we did not include this quantity for the sake of clarity, the efficiency of the attack must also be taken into account. For example, the related-key attacks against full-round AES-192 and AES-256 [BKN09, BK09] require the equivalent of at least 2^{100} encryptions which remains much higher than the key size of some lightweight block ciphers. Related-key attacks are further discussed in Section 5.3.1.

[7]The project CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) aims at identifying the best authenticated ciphers. All submissions are listed on the following webpage: http://competitions.cr.yp.to/caesar-submissions.html.

Table 6: A summary of all lightweight block ciphers published at cryptography and security conference we are aware of, sorted by publication date. The percentage of rounds attacked in the single key (SK) or related key (RK) model is the maximum over all versions of the cipher. If the cipher designers explicitly disregarded related key attacks then "–" is used. RK cannot be smaller than SK.

Description		Parameters			Fraction of Rounds Attacked		
Name	Ref.	Key	Block	Rounds	SK	RK	Ref.
3-Way	[DGV94]	96	96	11	0.45	**1**	[DGV94, KSW97]
RC5	[Riv95]	0–2040	32/64/128	12	**1**	**1**	[BK98]
TEA	[WN95]	128	64	32	0.53	**1**	[CWP12, KSW97]
Misty1	[Mat97]	128	64	8	**1**	**1**	[Tod17, BOK16]
TREYFER	[Yuv97]	64	64	32	**1**	**1**	[BW99]
XTEA	[NW97]	128	64	64	0.36	0.56	[SMVP11, Lu09]
AES	[DR98]	128/192/256	128	10/12/14	0.70	**1**	[DFJ13, BK09]
BKSQ	[DR00]	96	96/144/192	10/14/18	0.60	0.60	[DR00]
Khazad	[BR00]	128	64	8	0.75	**1**	[BKP16, BN10]
Noekeon	[DPVAR00]	128	128	16	0.75	–	[DPVAR00]
Iceberg	[SPR+04]	128	64	16	0.50	0.50	[SWJS12]
mCrypton	[LK06]	64/96/128	64	12	0.83	0.83	[DF16]
HIGHT	[HSH+06]	128	64	32	0.84	**1**	[WWBC14, KHK11]
SEA	[SPGQ06]	96	96	≥ 105	$\emptyset$	$\emptyset$	$\emptyset$
CLEFIA	[SSA+07]	128/192/256	128	18/22/26	0.72	0.72	[MDS11]
DESLX	[LPPS07]	184	64	16	$\emptyset$	–	$\emptyset$
PRESENT	[BKL+07]	80/128	64	31	0.87	0.87	[BN14]
MIBS	[ISSK09]	64/80	64	32	0.59	0.59	[BHV14]
KATAN	[CDK09]	80	32/48/64	254	0.60	0.60	[FM15]
KTANTAN	[CDK09]	80	32/48/64	254	**1**	**1**	[BR11]
PRINTCipher	[KLPR10]	48/96	80/160	48/96	**1**	**1**	[LAAZ11]
EPCBC	[YKPH11]	96	48/96	32	**1**	**1**	[Bul13]
KLEIN	[GNL11]	64/80/96	64	12/16/20	**1**	**1**	[LN15]
LBlock	[WZ11]	80	64	32	0.75	0.75	[WWJ16]
LED	[GPPR11]	64/128	64	32/48	0.66	0.66	[DDKS16]
Piccolo	[SIH+11]	80/128	64	25/31	0.68	0.68	[SIH+11, Min13]
PICARO	[PRC12]	128	128	12	$\emptyset$	**1**	[CLNP16]
PRINCE	[BCG+12]	128	64	12	0.83	–	[CFG+15, DP15]
ITUbee	[KDH13]	80	80	20	0.40	0.40	[Sol15]
TWINE	[SMMK13]	80/128	64	36	0.69	0.75	[BDP15, BBR+13]
Zorro	[GGNS13]	128	128	24	1.0	1.0	[BDD+15]
PRIDE	[ADK+14]	128	64	20	0.90	0.90	[LR17]
Joltik † ‡	[JNP14a]	64/80/96/128	64	24/32	0.75	0.75	[JNP14a]
LEA	[HLK+14]	128/192/256	128	24/28/32	0.58	0.58	[SHY16]
iScream † ‡	[GLS+14]	128	128	12/14	**1**	**1**	[LMR15, TLS16]
LBlock-s †	[ZWW+14]	80	64	16/32	**1**	**1**	[Leu16a]
Scream † ‡	[GLS+14]	128	128	10/12	**1**	**1**	[TLS16]
Lilliput	[BFMT15]	80	64	30	0.57	0.57	[ST17]
RECTANGLE	[ZBL+15]	80/128	64	25	0.72	0.72	[ZBL+15]
Fantomas	[GLSV15]	128	128	12	$\emptyset$	–	$\emptyset$
Robin	[GLSV15]	128	128	16	**1**	–	[LMR15]
Midori	[BBI+15]	128	64/128	16/20	**1**	**1**	[TLS16]
SIMECK	[YZS+15]	64/96/128	32/48/64	32/36/44	0.80	0.80	[QHS17]
RoadRunneR	[BŞ16]	80/128	64	10/12	0.60	–	[BŞ16]
FLY	[KG16]	128	64	20	$\emptyset$	$\emptyset$	$\emptyset$
Mantis ‡	[BJK+16]	128	64	14	0.71	–	[DEKM16]
SKINNY ‡	[BJK+16]	64–384	64/128	32–56	0.59	0.59	[ABC+17, LGS17]
SPARX	[DPU+16]	128/256	64/128	24–40	0.69	0.69	[DPU+16, ATY17]
Mysterion	[JSV17]	128/256	128/256	12	$\emptyset$	–	$\emptyset$
GIFT	[BPP+17]	128	64/128	28/40	0.54	–	[BPP+17]
Qarma ‡	[Ava17]	128/256	64/128	16/24	$\emptyset$	–	$\emptyset$

3.2.4 Dedicated Authenticated Encryption Schemes

Following the call for submissions of the CAESAR competition, several lightweight authenticated encryption schemes were proposed. Some of them rely on dedicated block ciphers used with specific modes, in which case their underlying block cipher is in Table 6: iScream, Scream and Joltik. LAC uses a more sophisticated mode using the key schedule of its internal block cipher, LBlock-s, which is also in Table 6. However, other algorithms based on sponge transformation or stream cipher-like construction were also proposed. These are listed in Table 7. As with hash functions, sponge-based algorithms are marked with the symbol "‡" and algorithms for which a full-round attack exists are marked with "†".

Table 7: A summary of all lightweight authenticated ciphers from academia, sorted by publication date. The key, initialization vector (IV) and internal state (IS) are expressed in bits.

Name	Reference	Key	IV	IS
Helix	[FWS+03]	256	128	160
ASC-1	[JK12]	256	56	384
Hummingbird-2 †	[ESSS12]	128	64	128
ALE †	[BMR+14]	128	128	128
FIDES †	[BBK+13]	80/96	160/192	80/96
LAC †	[ZWW+14]	80	64	144
Sablier †	[ZSX+14]	80	80	208
TriviA	[CCHN15]	128	128	384
ACORN	[Wu16]	128	128	293
Ascon ‡	[DEMS16]	96/128	96/128	320
KETJE ‡	[BDP+16]	$\leq 182/382$	$182-k/382-k$	200/400
NORX32 ‡	[AJN16]	128	128	512

Hummingbird-2 was designed after Saarinen found attacks against the first version of this algorithm [Saa11]. Still, there exists a differential attack against full Hummingbird-2 in the related-key setting [Saa14]. Full-round ALE is vulnerable to the "LOCAL" attack [KR14]. FIDES was also broken shortly after its publication by Dinur and Jean [DJ15]. A nonce misuse allows a differential attack against Helix [Mul04]. The same paper also presents internal collisions. Leurent found a differential forgery attack against LAC [Leu16a]. Sablier was cryptanalysed by Feng and Zhang [FZ14]. Finally, full-round NORX v2 was found to be vulnerable to both key recovery and forgery attacks [CFG+17]. The current version, v3, was patched to prevent them.

3.2.5 Dedicated MACs

While far less popular than other structures, two lightweight message authentication codes have been proposed. They are listed in Table 8. Both are sponge-based, although the mode of operation used in SipHash is a bit more involved.

Table 8: A summary of all lightweight MACs from academia we are aware of. The key and internal state (IS) are expressed in bits.

Name	Reference	Key	Block	IS
SipHash ‡	[AB12]	64	64	256
Chaskey †	[MMH+14]	128	128	128

The designers of Chaskey proposed using the 128-bit internal permutation of their algorithm to build a block cipher using the Even-Mansour construction. They in fact

submitted the corresponding block cipher to the FELICS competition. Following a differential-linear attack against 7 out of 8 rounds of Chaskey [Leu16b], its designers encourage the use of 12- and 16-round versions of its internal permutation respectively called Chaskey-12 and Chaskey-LTS [Mou15].

3.3 Algorithms from Government Agencies

Governmental agencies have published their own lightweight ciphers. The publication is often done via national standards. These ciphers are usually targeting local usage but, due to the interconnection of the markets and the corresponding standardizing efforts, these can end up being used outside of their expected zone of influence.

The rationale behind the design of the NSA block cipher Skipjack is not known. The only public information about its engineering comes from the report written by external cryptographers after two days spent at the NSA headquarters [BDK+93] and from several attempts at reverse-engineering both its design process [KRW99, KW01] and its S-Box [BP15]. The same can be said of the Russian lightweight block cipher Magma specified in the latest Russian standard for block ciphers, GOST R 34.12–2015 [Fed15], which describes both this algorithm and the heavier Kuznyechik. Both were designed by the Russian secret service (FSB). Magma is based on a 64-bit 32-round block cipher designed much earlier which was referred to as "GOST block cipher" in the literature. It was later described in an RFC [Dol10]. A specific version of this older cipher called "GOST revisited" was described by independent researchers in 2010 [PLW10]. As the S-Boxes used by the original algorithms are not specified, Poschmann et al. suggested and benchmarked a version of this algorithm using the S-Box of PRESENT.

Other algorithms are in a slightly different situation: while the initial design was kept entirely secret, some information was eventually released by the designers. For example, design criteria used to build the S-Boxes of the DES [U.S99] have eventually been published [Cop94], including the fact that the NSA improved their resilience to differential attacks.

Also from the NSA, SIMON and SPECK are two block ciphers which were published on `eprint.iacr.org` [BSS+13] and are being considered for inclusion in the ISO/IEC standard for "lightweight cryptography" (see Section 4.1). The designers of these algorithms were invited to present these algorithms to the Design Automation Conference (DAC) of 2015 but their paper [BTCS+15] provides little insight into their design process. Following pressure from the academic community, its authors eventually published another note on `eprint.iacr.org` [BSS+17] where they provide more details about their goal. According to this document, their aim was to design ciphers with a security margin of 25-30% which about matches the best results on these ciphers in the public literature. However, this note does not describe any new attack; all the results about the ciphers are taken from the open literature. Incidentally, it provides a very comprehensive literature overview of the different attacks against the different versions of these algorithms.

The stream cipher ZUC was designed by the Data Assurance and Communication Security Research Center (DACAS) of the Chinese Academy of Science. It was published directly as part of the 3GPP standard [ETS11] using a structure reminiscent of that of SNOW 2.0. This addition was caused by the demand from the Chinese government to use Chinese algorithms when operating in China. ZUC should in theory not be used while in other countries. Again, some partial information is provided regarding its design. In particular, several modifications were made necessary by external cryptanalyses [WHN+12], as described in [ETS11]. Nevertheless, the cryptanalysis performed by its original designers is, to the best of our knowledge, still secret.

The public lightweight ciphers designed by government agencies are listed in Table 9. The stream cipher is marked with the "†" symbol. Attacks target full-round DES [Mat94] and full-round GOST, revisited or not [Iso13]; we thus gave them a "+" mark. Skipjack is

Table 9: A summary of all ciphers from government agencies. The key, initialization vector (IV) and internal state (IS) sizes are expressed in bits. For block ciphers, the internal state corresponds to the block.

Name	Designers	Reference	Key	IS	Rounds	IV
DES †	NSA/IBM	[U.S99]	56	64	16	–
GOST (revisited) †	KGB, then revisited by academics	[Dol10, PLW10]	256	64	32	–
Magma	FSB	[Fed15]	256	64	32	–
SIMON	NSA	[BSS+13]	64–256	32–128	32–72	–
Skipjack	NSA	[U.S98]	80	64	32	–
SPECK	NSA	[BSS+13]	64–256	32–128	22–34	–
ZUC †	Chinese Academy of Sciences	[ETS11]	128	560	–	128

vulnerable to an impossible differential attack targeting 31 out of its 32 rounds [BBS99]. As stated above, all versions of SIMON and SPECK have a security margin around 30%. To the best of our knowledge, no cryptanalysis have broken ZUC after its modification.

4 Lightweight Cryptography in the Wild

Several standards are aimed at use cases overlapping with those of lightweight cryptography. These are listed in this section and summarized in Table 10. Section 4.1 deals with ISO/IEC standards, Section 4.2 with regional cryptographic ones and Section 4.3 with general purpose communication protocols run by low power devices.

Table 10: Standards and algorithms recommended by public institutions involving lightweight algorithms.

Type	Name	Lightweight algorithms standardized
ISO/IEC	29167	AES-128, PRESENT-80, Grain-128A
	29192-2	PRESENT, CLEFIA
	29192-3	Enocoro, Trivium
	29192-5	PHOTON, Lesamnta-LW, Spongent
	18033-3	AES, MISTY1, HIGHT
	18033-4	SNOW 2.0, Rabbit
Regional	FIPS 185 (USA)	Skipjack (now deprecated [BR15])
	FIPS 197 (USA)	AES
	NESSIE (EU)	AES, MISTY1
	eSTREAM portfolio (EU)	Grain, Trivium, Salsa20, MICKEY, Rabbit
	CRYPTREC (Japan)	AES
	TTA (South Korea)	HIGHT, LEA
	GOST R 34.12–2015 (Russia)	Magma
Protocols	GSM	A5/1, A5/2, A5/3 (KASUMI)
	3G	SNOW 3G, ZUC, AES, KASUMI, Keccak
	Bluetooth	E0, AES
	Bluetooth smart	AES
	WEP	RC4
	WPA	RC4
	WPA2	AES
	Lora Alliance	AES
	IEEE 802.15.4 (Zigbee)	AES

4.1 ISO/IEC cryptographic standards.

The International Organization for Standards (ISO) and the International Electrotechnical Commission (IEC) are tasked with issuing and maintaining standards regarding information and communication technology.

Three of their standards are particularly relevant for lightweight cryptography. The first is *ISO/IEC 29167: Information technology – Automatic identification and data capture techniques*, in particular parts 10, 11 and 13. Those deal with the symmetric ciphers that should be used for securing "air interface communications", that is, RFID tags. These parts describe respectively AES-128, PRESENT-80 and Grain-128A. Other parts deal with public key cryptography.

Another set of relevant ISO/IEC standards are those with number 29192 which deal specifically with "lightweight cryptography". The following algorithms are part of this series of standards: the block ciphers PRESENT and CLEFIA, the stream ciphers Trivium and Enocoro, and the hash functions PHOTON, Spongent and Lesamnta-LW. The criteria for algorithms to be considered for inclusion in this standard are listed in the following quote from Annex A of said standard.

> a) The security of the cryptographic mechanism. 80-bit security is considered to be the minimum security strength for lightweight cryptography. It is however recommended that at least 112-bit security be applied for systems that will require security for longer periods (refer to SD12 for security strength references, as the period of protection provided is determined by the security strength as well as the computing power of the adversary who wishes to break the algorithm.).
>
> b) The hardware implementation properties (for hardware targeted mechanisms). The chip area occupied by the cryptographic mechanism (reduced compared to existing ISO standards) and the energy consumption. (clear advantage over existing ISO standards, e.g. ISO/IEC 18033, ISO/IEC 9798, ISO/IEC 11770).
>
> c) The software implementation properties (for software targeted mechanisms). In particular, the code size and the required RAM size. (Less resource requirements compared to existing standards on the same platform are considered as potentially lightweight for software environments).
>
> d) The nature of any licensing issues affecting the cryptographic mechanism.
>
> e) The maturity of the cryptographic mechanism.
>
> f) The generality of the lightweight properties claimed for the cryptographic mechanism (i.e. the more independent the claimed lightweight property is from implementation in a specific technology, the better, as it will be useable by a wider audience).

At the time of writing, the block ciphers SIMON and SPECK designed by the NSA were being considered for inclusion in this standard.

Finally, standard 18033 describes "Encryption algorithms". Some of those can be considered lightweight, such as the block ciphers AES, MISTY1 and HIGHT (in 18033-3) and the stream ciphers SNOW 2.0 and Rabbit (in 18033-4).

4.2 Regional Cryptographic Standards

Several regional standards deal with cryptography in general and some of the algorithm specified in them can be considered to be lightweight. In the USA, cryptographic standards are handled by the National Institute for Standards and Technology (NIST) which famously

standardized the AES after an open competition. This institution is currently working towards a standard for lightweight cryptography, as explained in a detailed report [MBTM17]. Their intention is to agree upon several *profiles* corresponding to different algorithms, use cases and constraints. Then, possibly different algorithms will be standardized for use in each of these profiles. The legacy cipher Skipjack was a NIST standard but it is now deprecated.

The Japanese Cryptography Research and Evaluation Committees (CRYPTREC) maintains the "e-Government Recommended Ciphers List" which contains algorithms whose usage should be preferred. It was last updated in 2013 and contains only one lightweight algorithm: the AES. However, CRYPTREC is also working on lightweight cryptography: in March 2017, they published a comprehensive technical report [CRY17] describing use cases for lightweight algorithms, several such primitives and extensive benchmarkings.

The Telecommunications Technology Association of Korea (TTA) provides the relevant standards in South Korea. Both HIGHT and LEA have been standardized in TTAS.KO-12.0040/R1 and TTAS.KO-12.0223[8] respectively.

In Europe, the NESSIE project selected several block ciphers including the AES and MISTY1. Its failure to find good stream ciphers lead to the eSTREAM competition. At its end, a portfolio of stream ciphers was published. It is divided into two profiles, one software oriented and one hardware oriented. Several of those stream ciphers can be considered to be lightweight: Trivium, Grain, MICKEY, Rabbit and Salsa20. Much like the Japanese CRYPTREC, the aim of NESSIE and eSTREAM is not formally standardize algorithms but to recommend some.

Finally, the latest Russian standard for block ciphers contains the 64-bit block cipher Magma as an explicit lightweight counterpart of the bigger Kuznyechik.

4.3 Communication protocols

Several communication protocols specify a form of encryption which, given the nature of the devices running them, have to be lightweight. For example, cell phones are not nearly as powerful as computers, although Moore's law and modern smartphones complicate this picture.

The GSM and 3G networks deal with cell phone communication. They specify that communications should be encrypted using A5/1, A5/2, A5/3 (KASUMI in counter mode), SNOW 3G, ZUC or KASUMI, the latter being a variant of MISTY1. An update of 3GPP from 2014 [ETS14] adds functions based on Keccak-f[1600] to provide key generation and authentication.

Bluetooth connects devices over short distances. The original specification required the stream cipher E0 but it was later replaced by the AES. A more recent variant called "Bluetooth smart" aims at lower energy consumption. It also relies on the AES for its security.

Modern WiFi connections are secured using WPA or WPA2. The former uses RC4 while the latter moved on to the AES. The previous standard was WEP, which used RC4, but practical attacks exist against it [FMS01].

Several protocols have recently been proposed to connect wireless IoT devices to one another. The one put forward by the Lora Alliance uses the AES. The same is true for IEEE 802.15.4, which is used e.g. in Zigbee.

[8] Short descriptions in English and links to the specifications (in Korean) are available at http://www.tta.or.kr/English/new/standardization/eng_ttastddesc.jsp?stdno=TTAK.KO-12.0040/R1 for HIGHT and at http://www.tta.or.kr/English/new/standardization/eng_ttastddesc.jsp?stdno=TTAK.KO-12.0223 for LEA.

5 Trends in Lightweight Design

Lightweightness can be seen as a set of specific design constraints. These are tackled differently by different algorithms but some trends emerge when we look at the evolution of lightweight block ciphers. These are particularly visible on several fronts: the choice of the non-linear operations, linear layers, and the key schedule which are described respectively in Section 5.1, Section 5.2, and in 5.3. We discuss modes of operations in Section 5.4 and note in Section 5.5 that fewer ad hoc ciphers seem to be in use now than 15 years ago.

5.1 Non-Linear Operations

Non-linearity is a necessary property of any cryptographic primitive. It can be provided by S-Boxes or through the use of non-linear arithmetic operations. An advantage of S-Boxes is the simple security argument based for example on the wide trail strategy that they allow. S-Box-based algorithms can further be divided into two categories. The first specify them using Look-Up Tables (LUT) and the second uses bit-sliced algorithms. These descriptions loosely map to implementation strategies, although it is possible to find bit-sliced implementations of S-Box specified via their LUTs. As for arithmetic operations, only modular additions are considered here, that is, primitives following the ARX paradigm. Although other operations are sometimes used, such as modular multiplication in the block cipher IDEA [LM91] and the PC1 stream cipher, these are extremely uncommon.

5.1.1 Look-Up Table

LUT-*based* algorithms use S-Boxes which are specified via their look-up tables and which are usually implemented via this lethod in software. Such functions are useful as they can offer (near) optimal cryptographic properties using what is essentially a unique operation. However, such an implementation requires storing all possible outputs which, for an 8-bit S-Box such as the one used by the AES, has a significant cost. Furthermore, the table look-up is the operation leaking the most information, as shown in [BDG16].

S-Boxes intended to be implemented using LUTs in software usually correspond to a simple electronic circuit which can be efficiently implemented in hardware, such as the 4-bit S-Boxes used by Piccolo, PRESENT or PRINCE. In the case of hardware, it is possible to significantly reduce the cost of the S-Box of the AES because of its algebraic structure as famously done e.g. by Canright [Can05].

5.1.2 Bit-slice

Bit-slice-based algorithms also use S-Boxes but, in this case, the S-Box is intended from the start to be implemented in a bit-sliced fashion: no table look-ups are required to evaluate the S-Box layer. Instead, some bitwise operations such as AND and XOR are performed on words of w bits, thus evaluating the S-Box in parallel w times. It is possible to find bit-sliced implementations for S-Boxes specified via their LUT but, in the case of 8 bits S-Boxes, these are usually much more expensive that the bit-sliced implementation of S-Boxes designed from the start with this implementation strategy in mind.

S-Boxes specified via a bit-sliced representation require only a limited number of logical operations to be evaluated: 4-bit ones usually need only 4 non-linear gates[9] during their evaluation which makes their software implementation particularly easy to mask. A simple bit-sliced implementation is also related—but is not equivalent to—a small area for a hardware implementation, meaning that such algorithms can be expected to perform well in hardware as well.

[9] At least 4 are needed in order to avoid having linear coordinates. In practice, it turns out that using 4 is sufficient to have very good cryptographic properties.

Because of these properties, bit-sliced S-Boxes are a popular choice for the design of lightweight algorithms, especially during the last 4 years. For example, all the algorithms in the following list use such components.

- 3-Way
- Ascon
- Fantomas
- Fly
- iScream
- Ketje
- Mysterion
- Noekeon
- Pride
- Rectangle
- RoadRunneR
- Scream

5.1.3 ARX-based

Arx-*based* algorithms rely on modular addition to provide non-linearity while word-wise rotations and XOR provide diffusion, hence the name: Addition, Rotation, XOR.

The bits of highest weight in the output of a modular addition are highly non-linear functions due to the propagation of the carry. However, the lower weight bits retain a simple dependency. Furthermore, some differentials and linear approximations have probability 1, meaning that the structure of the linear part must be studied carefully. For example, care must be taken when choosing the rotation amounts. Justifying the security of an ARX-based algorithm is far more difficult than for an S-Box-based design because tools such as the wide-trail strategy cannot be applied. In fact, to the best of our knowledge, SPARX [DPU+16] is the only ARX-based primitive designed to be provably secure against differential and linear attacks.

Modular addition is extremely cheap in software. Not only does it consist in few operation,[10] it also uses fewer or no additional registers as it can often be performed in place using the "`+=`" operator.

As a consequence, ARX-based ciphers are among the best performers for micro-controllers identified using FELICS. Some ARX-based block and stream ciphers are listed below.

- Chacha20
- Chaskey
- Hight
- Lea
- RC5
- Salsa20
- SPARX
- Speck
- Tea
- Xtea

5.1.4 On SCA Counter-Measures

The choice of the non-linear layer is particularly important when it comes to easing the implementation of counter-measures against side-channel attacks. Indeed, depending on the structure of the non-linear layer, these counter-measure can be more or less costly and have a varying impact on performances.

Popular counter-measure against side-channel attacks are *threshold implementations* and *masking* which was briefly described in Section 2.3. Both rely on secret sharing: the idea is to obfuscate the operands of operations whose physical characteristics (e.g. their precise power consumption over time) depend on the value of their input. Threshold implementations must satisfy more properties than masked ones which allows them to provide provable security against some adversaries. Both require an external source of random bits which will come with its own cost. The aim in this context is to provide security guarantees against different forms of side-channel attacks while using as little random bits as possible and as little additional logic as possible. Note that the situation in software and hardware are fairly similar in this regard.

Particular attention has been paid to threshold implementation of the multiplicative inverse in $\mathrm{GF}(2^8)$ since it is the operation the S-Box of the AES is based on. It is possible

[10] In fact, if the operands are of the size of the register then only one operation is needed.

to leverage its strong algebraic structure to provide efficient threshold implementations of such components as done for example in [MPL+11] or, more recently, in [BGN+14].

If the algorithm uses an ARX structure, then it is possible to use the arithmetic structure of modular addition to mask this operation directly. Rather than masking each carry propagation bit separately, we can instead generate secret shares and combine them using modular addition directly. The problem is then in the interaction between the masking of the modular addition and the masking of the linear part because they are done in different groups. This topic has been investigated by multiple teams e.g. in [CG00, CGTV15].

The cost of a threshold implementation increases with the algebraic degree of the operation considered. As a consequence, the threshold implementation of a large and cryptographically strong S-Box is likely to be prohibitively expensive unless it was designed with a special structure. As discussed above, the algebraic structure of the S-Box of the AES can be used but simpler structure are possible. For example, the S-Box of the Keccak permutation operates on 5 bits using a simple bit-sliced approach and is also quadratic, meaning that each output bit has algebraic degree 2. This simplicity implies a low cost for theshold implementations, as shown for example in [BDN+14].

Such observations partially explain the popularity of bit-sliced S-Boxes: while their performances are already good in the non-protected case, they allow even better side-channel-resilient implementations. This was explicitly intended by the designers of the LS-designs [GLSV15] Robin and Fantomas. Since the S-Box is evaluated in a bit-sliced fashion in these block ciphers, it is possible to mask each `AND` separately and to perform the corresponding operations in parallel over all S-Boxes at the same time. However, in order for this method to be efficient, it is necessary that the S-Box uses as few `AND`s as possible, which is at odds with its cryptographic strength. Designers must therefore find a balance between strength against cryptanalysis and strength against side-channel attacks. It should be noted that this trade-off is mostly relevant for large S-Boxes. For instance, those operating on 4 bits can use only 4 non-linear gates while retaining an optimal security against linear and differential attacks.

5.2 Linear Operations

The role of the linear operations is to provide diffusion, i.e. to ensure that all parts of the state depend on one another after several rounds.

5.2.1 MDS Matrices

A Maximum Distance Separable (MDS) matrix M operating on vectors of length d is such that $\mathrm{hw}(x) + \mathrm{hw}(M(x)) \leq d + 1$ for all non-zero x, where $\mathrm{hw}(x)$ counts the number of non-zero elements in the vector x. Such matrices provide optimal diffusion and can be used to prove the resistance against linear and differential attacks of the ciphers built using them. This technique, the *wide trail argument*, was most prominently used in the design of the AES and was later used by many algorithms directly inspired from it. Other algorithms not based on the AES use such components as well. Such algorithms are listed below.

- AES
- CLEFIA
- LED
- Lesamnta-LW
- PHOTON
- SNOW-3G
- Zorro
- ...

Linear Feedback Shift Registers (LFSRs) operating on elements of $\mathrm{GF}(2^n)$ rather than bits can be used to build MDS matrices that are efficiently implemented by iterating a simple transformation several times. A comprehensive survey on this topic can be found in [JPS17].

While they provide optimal diffusion, such diffusion layers tend to be more expensive than the others. The common trade-off between performance and security level is present in this context as well.

5.2.2 Bit Permutations

Due to their very low cost in hardware, simple bit permutations are a popular choice for algorithms targeting such platforms. Most prominently, the DES and PRESENT use such linear layers. The downside is that they are a priori quite expensive in software which is usually ill-suited for efficient bit fiddling. The one exception to this general rule is a simple rotation of a word by a fixed amount. While such rotations may provide less diffusion than custom ones as used in EPCBC, PRESENT or GIFT, they allow a far more efficient implementation in software. This approach is used for example by FLY, RECTANGLE and RoadRunneR.

Generalized Feistel networks (GFN) relying entirely on nibble permutation to provide diffusion between their branches can also be fitted in this category. In their case, a software implementation is a bit cheaper because the bits are grouped into larger groups, meaning that it is not necessary to treat the bits one by one. On the other hand, diffusion can be fairly slow in the case of standard GFN, meaning that many rounds are necessary. HIGHT uses 32 rounds as a consequence of this structure. To mitigate this problem, nibble permutations offering faster diffusion than the usual rotation can be used such as those proposed by Suzaki and Minematsu in [SM10]. They built TWINE using such a permutation. Piccolo, LBlock, LBlock-s, and Lilliput are other GFNs using more elaborate branch permutations.

5.2.3 XOR and Rotations

To find a compromise between diffusion, cost in software, and cost in hardware, a popular choice is to build a linear layer using word-wise rotations along with word-wise XORs. Such a method has the advantage of allowing an efficient software implementation while remaining cheap in hardware since the rotations correspond to simple bit permutations. This approach is used by all the ARX designs as well as the following ones.

- 3-Way
- Ascon
- Blake2s/b
- GIMLI
- GLUON
- Hummingbird-2
- ITUbee
- Noekeon
- ZUC

5.3 Key Schedule

The key schedule is the area where lightweight algorithms differ the most from their non-lightweight counterparts. Indeed, for algorithms intended to run on standard computers, it is fine to have a complex key schedule as it would typically be run only once, the corresponding subkeys being subsequently stored. For lightweight algorithms, the incurred cost in terms of RAM or gate area is unacceptable. Furthermore, several lightweight algorithms dismiss resilience against related key attacks altogether, a design decision which authorizes the use of much simpler key schedules.

Different attitudes regarding related-key attacks are discussed in Section 5.3.1. Popular strategies for building simple key schedules are described in Sections 5.3.2 and 5.3.3. Some recent proposals provide a tweak in addition to the secret key [LRW02]. This additional parameter is discussed later in Section 5.4.

5.3.1 On Related-Key Attacks

Some cipher designers claim resilience against related-key attacks while some other algorithms are trivially vulnerable against such attacks. For example, the block cipher PRINCE has very simple related-key distinguishers because of its α-reflection and its FX constructions. On the other hand, other algorithms explicitly give resilience against related-key as a design criteria.

Preventing related-key attacks is a more conservative choice. Whatever the setting, from a security standpoint, being protected against such adversaries can only be an advantage. And yet this resilience has a cost since it implies the use of more rounds and/or more complex key schedules which lead to a performance degradation particularly unwelcome in the lightweight setting. Furthermore, for devices using a unique factory-defined key throughout their lifetimes, the probability of finding two devices with the appropriate key relation is small enough that it is of no practical concern. Similarly, if the protocols using the ciphers are properly implemented, related-key attacks should not be possible.

The algorithms in the list below were explicitly *not* designed to prevent related-key attacks. Still, the approach used for Noekeon and FLY is a bit more subtle. Indeed, for cases where related-key attacks might be of concern, the authors provide a modified key schedule. While normally the master key is simply XORed in the state, as for the ciphers in Section 5.3.2, the related-key protected version imposes that the master key first goes through several rounds of the round function so as to break any pattern relating the keys. Similarly, the authors of GIFT suggest doubling its number of rounds if related-key security is needed.

- Fantomas
- Fly
- Mysterion
- Noekeon
- Pride
- PRINCE
- Zorro
- ...

Some designers prefer to make the most conservative choice by providing related-key security. Some of the corresponding algorithms are listed below. These usually employ a more complex key-schedule but, since they remain lightweight ciphers, those can be evaluated "on the fly" cheaply. It means that the subkeys are obtained by extracting bits from a key state which is updated in every round, just like the internal state of the block cipher. However, this update function is kept simple to limit the performance overhead.

- Epcbc
- LBlock
- Sea
- Simon
- Skinny
- SPARX
- TWINE
- ...

5.3.2 Even-Mansour and "Selecting" Key Schedules

It is popular for lightweight algorithms to use a key schedule which merely selects different bits of the master key in each round for use as subkey material along with some round constants. If the master key of a block cipher is simply XORed to the internal state during each round along with a round constant, the key schedule can be seen as a variant of the Even-Mansour construction [EM97]. Of course, it is possible to use said construction directly, as is the case for the Chaskey cipher. It is also possible to use different chunks of the master key during encryption. For example, Skipjack uses a 32-bit subset of its 80-bit master key in each round. The subkeys therefore repeat themselves every 5 rounds. We call such a key schedule a *selecting* key schedule since it merely selects some bits of the master key for use as subkeys.

The main advantage of such methods is that they require very little logic to compute the round keys. Furthermore, they have no need for a key state getting updated at each round which would be particularly expensive in hardware. This observation is what lead the designers of the stream cipher Sprout, followed later by those of Lizard and Plantlet,

to fix the content of one of their registers to be the master key without modifying it. In fact, the paper introducing Plantlet [MAM17] provides a detailed analysis of the way a key stored in non-volatile memory can be accessed and its impact on both performance and algorithm design. For example, it is better to access master key bits sequentially, like in Skipjack and in LED, than to use master key bits that are far apart to build a given round key.

Below, we list all ciphers using the master key in such a way that no key state needs to be maintained. It encompasses the (iterated) Even-Mansour construction, the "selecting" key schedules and the stream ciphers that do not modify their key register. Stream ciphers are indicated by the "†" symbol.

- 3-Way
- Chaskey
- DES
- Fantomas
- FLY
- GOST revisited
- HIGHT
- iScream
- ITUbee
- KTANTAN
- LED
- Lizard †
- Magma
- Midori
- Mysterion
- Noekeon
- Piccolo
- Plantlet
- PRIDE
- PRINCE
- RoadRunneR
- Robin
- Scream
- Skipjack
- Sprout †
- XTEA
- Zorro

The impact of such a key schedule in terms of gate area in hardware is extensively discussed in the recent paper [MAM17] which introduced Plantlet.

5.3.3 Round Function Based

A simple strategy to have a substantial key schedule while minimising its cost is to reuse significant parts of the round function to update the key state. The whole round function can be used, as in SPECK, or only parts of it, as in SPARX. Several block ciphers using this principle are listed below. For FLY and Noekeon, only the key schedule protecting against related-key attacks is concerned.

- EPCBC
- FLY
- Noekeon
- SEA
- SIMECK
- SPARX
- SPECK

5.4 Modes of Operation

The focus of this paper is lightweight symmetric primitives but modes of operation can also be more or less suitable for constrained platforms. For example, given the cost of memory in hardware, smaller block sizes are common for lightweight block ciphers (see Table 6) A small block size can be a problem as the security of some modes of operation such as CBC erodes very quickly when the number of n-bit blocks encrypted approaches $2^{n/2}$, as exploited for example in [BL16]. As a consequence, it makes sense to consider dedicated modes of operation such as CENC [Iwa06], an encryption mode secure beyond the birthday bound.

Another option that still relies on block cipher is the addition of a tweak in the primitive. This can be done either by building a tweak into the primitive itself, using for example the TWEAKEY framework [JNP14b], or by combining regular block cipher using an appropriate mode as explained e.g. in [Men15, WGZ+16]. The *tweakable block cipher* obtained in this way can be plugged into special modes that offer beyond the birthday-bound security for a smaller cost. This approach is used by some CAESAR candidates such as Joltik and SCREAM/iSCREAM. However, the downside of tweakable block ciphers is that they require additional memory to store the tweak itself.

As is visible from both the list of authenticated ciphers in Table 7 and the list of hash functions in Table 5, the sponge construction is a popular choice for constructing lightweight algorithms. Let us see possible reasons why this is the case.

Consider a Merkle-Damgård-based hash function with a compression function based on a block cipher with an n-bit block and a k-bit key. It operates on $n + k$ bits to process each block while providing at most $n/2$ bits of resilience against collision search. However, k is usually not small as it very often corresponds to the key-length of a block cipher. Similarly, a sponge-based hash function uses a permutation operating on $r + c$ bits during the absorption of each r-bit block while providing at most $c/2$ bits of security against collision search. However, it is possible to have rate r as small as a single bit. In fact, using $r = 8$ is a popular choice for lightweight hash functions: versions of Quark, Gluon and Spongent use it. The sponge structure allows a better use of limited memory. While it would in theory be possible to build a compression function with a very small second input, we are not aware of it ever being done.

The sponge construction can also be used to provide other functionalities like authenticated encryption using somewhat simpler modes than not-tweakable block ciphers [BDPV12]. In fact, several CAESAR candidates are sponge-based such as Ascon and Ketje. All the sponge permutation used to build hash functions could be used to provide authenticated encryption and vice-versa, although the number of rounds of the permutation must be considered carefully as authenticated cipher tend to use fewer during the encryption phase. Table 11 lists all lightweight sponges we are aware of.

Table 11: A summary of all lightweight sponges.

Name	Reference	Internal state size (bits)
Keccak	[U.S15]	200/400/800/1600
Quark	[AHMN10]	136/176/256
Photon	[GPP11]	100/144/196/256/288
Spongent	[BKL+11]	88/136/176/240/272
Gluon	[BDM+12]	136/176/256
Spn-Hash	[CYK+12]	128/256
SipHash	[AB12]	256
Ascon	[DEMS16]	320
Chaskey	[MMH+14]	128
Norx32	[AJN16]	512
Gimli	[BKL+17]	384
sLiSCP	[ARH+17]	192/256

5.5 No More Non-Standard Ciphers?

So far, we have discussed trends regarding algorithm design. But there is another trend at a higher level: broken ciphers such as those listed in Section 3.1 are being phased out. Nowadays, the prevalence of the AES means that using algorithms such as A5/1 would be unacceptable. Not only new standards are concerned: previously existing standards such as Bluetooth have been amended to move away from their previous *ad hoc* solutions (here, the E0 stream cipher) and towards more common choices (for Bluetooth, the AES). The reason behind this change is probably two-fold.

First, the lessons from the attacks targeting proprietary algorithms have likely been leaned. Thus, once these standards had to be replaced by more modern ones, the cryptography used was updated at the same time.

Second, the qualities of the AES likely played a significant role. The fact that it performs decently on a wide variety of platforms means that it a priori constitutes a satisfactory choice in a lot of situations. As a consequence of this versatility and of its

resilience to cryptanalysis, the AES has been formally standardized for use in most areas, as explained in Section 4. Still, the time it took for the algorithms from Section 3.1 to be phased out shows the importance of getting an algorithm choice right from the start.

6 Trade-Offs in Lightweight Cryptography

In this section, we discuss some of the higher level trade-offs at the core of lightweight cryptography: performance versus security (Section 6.1) and specialization versus versatility (Section 6.2). In Section 6.3, we argue that these two trade-offs are far from independent and that lightweight algorithms can be sorted in two broad categories as a consequence.

6.1 Performances vs. Security

Good performances and good security are at odds against one another in multiple ways. While this trade-off is not specific to lightweight cryptography, it is more pressing in this context because of the greater performance constraints that occur in this case.

The simplest is the number of rounds. Using fewer leads to a lower latency and, depending on the implementation, to a higher throughput. However, it also leads to a decreased security margin and may in fact lead to cryptanalysis. For example, LBlock-s as used in the CAESAR candidate LAC has only 16 rounds in most cases. While the full 32-round cipher seems safe, the usage of the 16-round version lead to differential forgery attacks [Leu16a]. Overall, choosing the right number of rounds for a primitive requires a difficult balance between speed and security.

This trade-off manifests itself in subtler ways as well. For example, using a strong S-Box with a high algebraic degree, low differential uniformity and low linearity would increase the resilience of the algorithm against common attacks. However, such components are more expensive to implement, meaning that they are not really available. Furthermore, such good properties imply the use of a more complex function with a higher algebraic degree which is far harder to mask. As a consequence, many lightweight algorithms prefer to trade a strong S-Box for a weaker whose masked implementation is far more efficient. This is for example the design choice made for the LS-designs Robin, Fantomas, Scream and iScream.

6.2 Specialization vs. Versatility

A trade-off more specific to lightweight cryptography is that of specialization versus versatility. Should an algorithm be highly optimized from the ground up for a specific use case or should it instead try to accommodate different niches of the design spaces as best as possible at the likely cost of not excelling in either context?

Primitives have been designed at different extremes of this trade-off. For example, KTANTAN, QUARK and Plantlet have been optimized for a low gate count in hardware implementation. On the other hand, the huge number of rounds required by the simplicity of their round function mean that a low latency implementation would be very difficult to design. On the other hand, the low latency block cipher PRINCE uses a more complex round function which uses different binary matrices for different parts of the internal state which would complicate a serialized implementation. In fact, the rounds themselves have different structures. A very small serialized implementation would thus be difficult.

These are but some of the many examples of highly dedicated algorithms that are listed in this paper, i.e. algorithms that have been designed for a specific use case on a specific platform, which excel in this context but which can be more challenging to implement efficiently outside of their “comfort zone”.

This specialization also occurs at the functional level. The α-reflection implemented by PRINCE means that a hash function built using a compression function based on PRINCE in a Merkle-Damgård structure is likely to have significant problems. Similarly, stream ciphers cannot easily be plugged into modes of operations to build other primitives like hash functions or authenticated ciphers.

On the other hand, other algorithms have been explicitly designed to provide good performances across many different platforms. For example, the block cipher Noekon uses a bit-sliced S-Box which is also easy to implement in hardware. Its linear layer relies on 32-bit word operations, namely rotations and XORs, so that it can be efficiently implemented both in software and hardware. Furthermore, some of the rotations are by multiples of 8 to further cheapen its implementation on 8-bit micro-controllers.

More generally, the strategy consisting in relying only on word-wide logical operations and rotations provides a good trade-off in terms of efficiency across multiple platforms. It is the reason why the recent GIMLI sponge, presented at CHES'17, uses such operations. Its designers explicitly targeted an efficient implementation on pretty much all platforms, from masked ASICs to 32-bit processors. As discussed in Section 5.4, the choice of a sponge permutation also implies functional versatility: unlike a stream cipher which can only provide encryption, a sponge can provide hashing, encryption and even authenticated encryption very simply. Thus, if versatility is the aim, it makes sense to design a sponge.

6.3 Subsets of Lightweight Cryptography

The two trade-offs described in Sections 6.1 and 6.2 are not independent. Algorithms for which a lower security level was deemed sufficient by the designers—such as an 80-bit key—tend to be designed with a specific platform and use case in mind. On the other hand, more generalist algorithms like Noekeon, GIMLI or Ascon make comparatively more conservative choices: Noekeon uses a 128-bit key while the internal states of GIMLI and Ascon (respectively 384 bits and 320 bits) are bigger than those of sponges designed for hardware like the one of QUARK (between 136 and 256 bits) or PHOTON (between 100 and 288 bits).

We thus see two subsets of algorithms, some being very specialized and willingly making bolder design choices to improve performances while others strive for versatility and more conservative security margins. In fact, the AES is a good example of the latter. While its smallest implementations [FWR05, BBR16b] are bigger than those of hardware oriented stream ciphers like Grain, they remain fairly competitive; and, unlike stream ciphers, it can be used to encrypt, authenticate, etc. In light of this, we propose splitting the primitives into the following two categories and, more importantly, suggest that designers use this distinction to more precisely classify their algorithms.

Ultra-Lightweight Cryptography corresponds to algorithm that fit in very specific areas of the design space such as "stream cipher with low gate area in ASICs" or "block cipher with high speed on 32-bit micro-controllers". As performance strongly dictates the use of such algorithms, making bolder choices in the performance versus security trade-off makes sense. Furthermore, as the use case is well specified, providing only one functionality such as encryption or authentication is not a problem. Examples of primitives that could fit in this category include, among others: Grain (stream cipher with low gate count in hardware), Qarma (tweakable block cipher with low latency in hardware), and Chaskey (MAC with high speed on micro-controllers).

Ubiquitous Cryptography deals with primitives that are geared towards versatility in terms both of functionality and implementation properties. They should efficiently run on a wide variety of platforms (8-, 16-, 32-bit micro-controllers, ASICs, FPGAs, 32- and 64-bit processors) and allow an efficient implementation of counter-measures against side-channel attacks. They should also provide various functions: hashing,

encryption, authentication, etc. Ascon, GIMLI and indeed the AES fit in this category.

7 Conclusion

Lightweight cryptography has received significant attention in the last two decades and even more so in the last 5 years. The AES, while very versatile both in software and in hardware, cannot address all design constraints. The need for lightweight algorithms is well established, as is evidenced by the NIST and CRYPTREC work to select and standardize such algorithms.

In this paper we make our best effort for an unbiased and comprehensive survey of the state-of-the art of the lightweight cryptography research field. Even given the range of opinions on what is lightweight and how to proceed with standardization in this area, we hope that this work can be a useful starting point helping to find common ground for researchers in the field and a starting reference point for young cryptographers. Finally we suggest a distinction between *ultra-lightweight* and *ubiquitous* cryptography which will hopefully provide a useful guideline in particular when discussing the level of security which a primitive should provide.

8 Acknowledgement

The authors thank Daniel Dinu and Yann le Corre for fruitful discussions about implementation and side-channel protection issues. We also thank ToSC reviewers, Nicky Mouha and NIST cryptographers for their very useful and detailed feedback on a previous version of this paper. The work of Léo Perrin was partially supported by the CORE project ACRYPT (ID C12-15-4009992) funded by the Fonds National de la Recherche, Luxembourg.

References

[AB12] Jean-Philippe Aumasson and Daniel J. Bernstein. SipHash: A fast short-input PRF. In Steven D. Galbraith and Mridul Nandi, editors, *Progress in Cryptology - INDOCRYPT 2012: 13th International Conference in Cryptology in India*, volume 7668 of *Lecture Notes in Computer Science*, pages 489–508. Springer, Heidelberg, December 2012.

[ABC+17] Ralph Ankele, Subhadeep Banik, Avik Chakraborti, Eik List, Florian Mendel, Siang Meng Sim, and Gaoli Wang. Related-key impossible-differential attack on reduced-round skinny. In Dieter Gollmann, Atsuko Miyaji, and Hiroaki Kikuchi, editors, *ACNS 17: 15th International Conference on Applied Cryptography and Network Security*, volume 10355 of *Lecture Notes in Computer Science*, pages 208–228. Springer, Heidelberg, July 2017.

[ABL+09] François Arnault, Thierry P. Berger, Cédric Lauradoux, Marine Minier, and Benjamin Pousse. A new approach for FCSRs. In Michael J. Jacobson Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *SAC 2009: 16th Annual International Workshop on Selected Areas in Cryptography*, volume 5867 of *Lecture Notes in Computer Science*, pages 433–448. Springer, Heidelberg, August 2009.

[ABN+11] Mohamed Ahmed Abdelraheem, Céline Blondeau, María Naya-Plasencia, Marion Videau, and Erik Zenner. Cryptanalysis of ARMADILLO2. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology –*

ASIACRYPT 2011, volume 7073 of *Lecture Notes in Computer Science*, pages 308–326. Springer, Heidelberg, December 2011.

[ABZD11] Mohamed Ahmed Abdelraheem, Julia Borghoff, Erik Zenner, and Mathieu David. Cryptanalysis of the light-weight cipher a2u2. In Liqun Chen, editor, *Cryptography and Coding: 13th IMA International Conference, IMACC 2011, Oxford, UK, December 12-15, 2011. Proceedings*, volume 7089 of *Lecture Notes in Computer Science*, pages 375–390, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[ADK+14] Martin R. Albrecht, Benedikt Driessen, Elif Bilge Kavun, Gregor Leander, Christof Paar, and Tolga Yalçin. Block ciphers - focus on the linear layer (feat. PRIDE). In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 57–76. Springer, Heidelberg, August 2014.

[AHMN10] Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and María Naya-Plasencia. Quark: A lightweight hash. In Mangard and Standaert [MS10], pages 1–15.

[AJN16] Jean-Philippe Aumasson, Philipp Jovanovic, and Samuel Neves. NORX v3.0. Candidate for the CAESAR Competition. See also https://norx.io, 2016.

[AM15] Frederik Armknecht and Vasily Mikhalev. On lightweight stream ciphers with shorter internal states. In Leander [Lea15], pages 451–470.

[And94] Ross Anderson. A5 (Was: HACKING DIGITAL PHONES). uk.telecom (Usenet), https://groups.google.com/forum/?msg/uk.telecom/TkdCaytoeU4/Mroy719hdroJ#!msg/uk.telecom/TkdCaytoeU4/Mroy719hdroJ, June 1994.

[ARH+17] Riham AlTawy, Raghvendra Rohit, Morgan He, Kalikinkar Mandal, Gangqiang Yang, and Guang Gong. sLiSCP: Simeck-based permutations for lightweight sponge cryptographic primitivesk. In Carlisle Adams and Jan Camenisch, editors, *Selected Areas in Cryptography – SAC 2017: 24th International Conference, Revised Selected Papers*, Lecture Notes in Computer Science, Cham, 2017. Springer International Publishing. to appear (available at https://eprint.iacr.org/2017/747).

[ATY17] Ahmed Abdelkhalek, Mohamed Tolba, and Amr M. Youssef. Impossible differential attack on reduced round SPARX-64/128. In Marc Joye and Abderrahmane Nitaj, editors, *AFRICACRYPT 17: 9th International Conference on Cryptology in Africa*, volume 10239 of *Lecture Notes in Computer Science*, pages 135–146. Springer, Heidelberg, May 2017.

[Ava17] Roberto Avanzi. The QARMA block cipher family. almost MDS matrices over rings with zero divisors, nearly symmetric Even-Mansour constructions with non-involutory central rounds, and search heuristics for low-latency S-Boxes. *IACR Transactions on Symmetric Cryptology*, 2017(1):4–44, 2017.

[BBI+15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 411–436. Springer, Heidelberg, November / December 2015.

[BBK03] Elad Barkan, Eli Biham, and Nathan Keller. Instant ciphertext-only cryptanalysis of GSM encrypted communication. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 600–616. Springer, Heidelberg, August 2003.

[BBK08] Elad Barkan, Eli Biham, and Nathan Keller. Instant ciphertext-only cryptanalysis of GSM encrypted communication. *Journal of Cryptology*, 21(3):392–429, July 2008.

[BBK+13] Begül Bilgin, Andrey Bogdanov, Miroslav Knežević, Florian Mendel, and Qingju Wang. Fides: Lightweight authenticated cipher with side-channel resistance for constrained hardware. In Bertoni and Coron [BC13], pages 142–158.

[BBR+13] Andrey Bogdanov, Christina Boura, Vincent Rijmen, Meiqin Wang, Long Wen, and Jingyuan Zhao. Key difference invariant bias in block ciphers. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 357–376. Springer, Heidelberg, December 2013.

[BBR16a] Subhadeep Banik, Andrey Bogdanov, and Francesco Regazzoni. Atomic-AES: A compact implementation of the AES encryption/decryption core. In Orr Dunkelman and Somitra Kumar Sanadhya, editors, *Progress in Cryptology – INDOCRYPT 2016*, volume 10095 of *Lecture Notes in Computer Science*, pages 173–190, Cham, 2016. Springer International Publishing.

[BBR16b] Subhadeep Banik, Andrey Bogdanov, and Francesco Regazzoni. Atomic-AES: A compact implementation of the AES encryption/decryption core. In Orr Dunkelman and Somitra Kumar Sanadhya, editors, *Progress in Cryptology - INDOCRYPT 2016: 17th International Conference in Cryptology in India*, volume 10095 of *Lecture Notes in Computer Science*, pages 173–190. Springer, Heidelberg, December 2016.

[BBS99] Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 12–23. Springer, Heidelberg, May 1999.

[BC13] Guido Bertoni and Jean-Sébastien Coron, editors. *Cryptographic Hardware and Embedded Systems – CHES 2013*, volume 8086 of *Lecture Notes in Computer Science*. Springer, Heidelberg, August 2013.

[BCG+12] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knežević, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In Wang and Sako [WS12], pages 208–225.

[BD04] M. Becker and A. Desoky. A study of the DVD content scrambling system (CSS) algorithm. In *Proceedings of the Fourth IEEE International Symposium on Signal Processing and Information Technology, 2004.*, pages 353–356, Dec 2004.

[BD08] Steve Babbage and Matthew Dodd. The MICKEY stream ciphers. In Matthew Robshaw and Olivier Billet, editors, *New Stream Cipher Designs*, volume 4986 of *Lecture Notes in Computer Science*, pages 191–209. Springer Berlin Heidelberg, 2008.

[BDD+15] Achiya Bar-On, Itai Dinur, Orr Dunkelman, Virginie Lallemand, Nathan Keller, and Boaz Tsaban. Cryptanalysis of SP networks with partial non-linear layers. In Oswald and Fischlin [OF15], pages 315–342.

[BDE+13] Lejla Batina, Amitabh Das, Barış Ege, Elif Bilge Kavun, Nele Mentens, Christof Paar, Ingrid Verbauwhede, and Tolga Yalçın. Dietary recommendations for lightweight block ciphers: Power, energy and area analysis of recently developed architectures. In Michael Hutter and Jörn-Marc Schmidt, editors, *Radio Frequency Identification: Security and Privacy Issues 9th International Workshop, RFIDsec 2013, Graz, Austria, July 9-11, 2013, Revised Selected Papers*, volume 8262 of *Lecture Notes in Computer Science*, pages 103–112, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[BDG16] Alex Biryukov, Daniel Dinu, and Johann Großschädl. Correlation power analysis of lightweight block ciphers: From theory to practice. In *International Conference on Applied Cryptography and Network Security – ACNS 2016*, volume 9696 of *Lecture Notes in Computer Science*, pages 537–557. Springer, 2016.

[BDK+93] Ernest F. Brickell, Dorothy E. Denning, Stephen T. Kent, David P. Maher, and Walter Tuchman. SKIPJACK review: Interim report. This note is available at http://faculty.nps.edu/dedennin/publications/SkipjackReview.txt., 1993.

[BDM+12] Thierry P. Berger, Joffrey D'Hayer, Kevin Marquet, Marine Minier, and Gaël Thomas. The GLUON family: A lightweight hash function family based on FCSRs. In Mitrokotsa and Vaudenay [MV12b], pages 306–323.

[BDN+10] Stéphane Badel, Nilay Dagtekin, Jorge Nakahara, Khaled Ouafi, Nicolas Reffé, Pouyan Sepehrdad, Petr Susil, and Serge Vaudenay. ARMADILLO: A multi-purpose cryptographic primitive dedicated to hardware. In Mangard and Standaert [MS10], pages 398–412.

[BDN+14] Begül Bilgin, Joan Daemen, Ventzislav Nikov, Svetla Nikova, Vincent Rijmen, and Gilles Van Assche. Efficient and first-order dpa resistant implementations of keccak. In Aurélien Francillon and Pankaj Rohatgi, editors, *Smart Card Research and Advanced Applications: 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, volume 8419 of *Lecture Notes in Computer Science*, pages 187–199, Cham, 2014. Springer International Publishing.

[BDP15] Alex Biryukov, Patrick Derbez, and Léo Perrin. Differential analysis and meet-in-the-middle attack against round-reduced TWINE. In Leander [Lea15], pages 3–27.

[BDP+16] Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Caesar submission: Ketje v2. Candidate for the CAESAR Competition. See also http://ketje.noekeon.org/, 2016.

[BDPV12] Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. Duplexing the sponge: Single-pass authenticated encryption and other applications. In Miri and Vaudenay [MV12a], pages 320–337.

[Ber08a] Daniel J. Bernstein. Chacha, a variant of Salsa20. SASC 2008 – the State of the Art in Stream Ciphers. See also https://cr.yp.to/chacha.html, 2008.

[Ber08b] Daniel J. Bernstein. The salsa20 family of stream ciphers. In Matthew Robshaw and Olivier Billet, editors, *New Stream Cipher Designs: The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 84–97, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[BFMT15] Thierry Pierre Berger, Julien Francq, Marine Minier, and Gaël Thomas. Extended generalized feistel networks using matrix representation to propose a new lightweight block cipher: Lilliput. *IEEE Transactions on Computers*, PP(99), August 2015.

[BGM06] Côme Berbain, Henri Gilbert, and Alexander Maximov. Cryptanalysis of Grain. In Robshaw [Rob06], pages 15–29.

[BGN+14] Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. A more efficient AES threshold implementation. In David Pointcheval and Damien Vergnaud, editors, *AFRICACRYPT 14: 7th International Conference on Cryptology in Africa*, volume 8469 of *Lecture Notes in Computer Science*, pages 267–284. Springer, Heidelberg, May 2014.

[BGS+05] Stephen C. Bono, Matthew Green, Adam Stubblefield, Ari Juels, Aviel D. Rubin, and Michael Szydlo. Security analysis of a cryptographically-enabled RFID device. In *Proceedings of the 14th Conference on USENIX Security Symposium - Volume 14*, SSYM'05, pages 1–1, Berkeley, CA, USA, 2005. USENIX Association.

[BHV14] Asli Bay, Jialin Huang, and Serge Vaudenay. Improved linear cryptanalysis of reduced-round MIBS. In Maki Yoshida and Koichi Mouri, editors, *IWSEC 14: 9th International Workshop on Security, Advances in Information and Computer Security*, volume 8639 of *Lecture Notes in Computer Science*, pages 204–220. Springer, Heidelberg, August 2014.

[Bih97] Eli Biham, editor. *Fast Software Encryption – FSE'97*, volume 1267 of *Lecture Notes in Computer Science*. Springer, Heidelberg, January 1997.

[Bir07] Alex Biryukov, editor. *Fast Software Encryption – FSE 2007*, volume 4593 of *Lecture Notes in Computer Science*. Springer, Heidelberg, March 2007.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Robshaw and Katz [RK16], pages 123–153.

[BJM+14] Lejla Batina, Domagoj Jakobovic, Nele Mentens, Stjepan Picek, Antonio De La Piedra, and Dominik Sisejkovic. S-box pipelining using genetic algorithms for high-throughput AES implementations: How fast can we go? In Willi Meier and Debdeep Mukhopadhyay, editors, *Progress in Cryptology - INDOCRYPT 2014: 15th International Conference in Cryptology in India*, volume 8885 of *Lecture Notes in Computer Science*, pages 322–337. Springer, Heidelberg, December 2014.

[BK98] Alex Biryukov and Eyal Kushilevitz. Improved cryptanalysis of RC5. In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 85–99. Springer, Heidelberg, May / June 1998.

[BK09] Alex Biryukov and Dmitry Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer, Heidelberg, December 2009.

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems – CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, Heidelberg, September 2007.

[BKL+11] Andrey Bogdanov, Miroslav Knežević, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. Spongent: A lightweight hash function. In Preneel and Takagi [PT11], pages 312–325.

[BKL+17] Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. GIMLI: A cross-platform permutation. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems – CHES 2017: 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 299–320, Cham, 2017. Springer International Publishing.

[BKLM09] Julia Borghoff, Lars R. Knudsen, Gregor Leander, and Krystian Matusiewicz. Cryptanalysis of C2. In Halevi [Hal09], pages 250–266.

[BKN09] Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic. Distinguisher and related-key attack on the full AES-256. In Halevi [Hal09], pages 231–249.

[BKP16] Alex Biryukov, Dmitry Khovratovich, and Léo Perrin. Multiset-algebraic cryptanalysis of reduced Kuznyechik, Khazad, and secret SPNs. *IACR Transactions on Symmetric Cryptology*, 2016(2):226–247, 2016. http://tosc.iacr.org/index.php/ToSC/article/view/572.

[BKZ11] Alex Biryukov, Ilya Kizhvatov, and Bin Zhang. Cryptanalysis of the Atmel cipher in SecureMemory, CryptoMemory and CryptoRF. In Lopez and Tsudik [LT11], pages 91–109.

[BL16] Karthikeyan Bhargavan and Gaëtan Leurent. On the practical (in-)security of 64-bit block ciphers: Collision attacks on HTTP over TLS and OpenVPN. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 456–467, New York, NY, USA, 2016. ACM.

[BLN15] Daniel J. Bernstein, Tanja Lange, and Ruben Niederhagen. Dual EC: A standardized back door. Cryptology ePrint Archive, Report 2015/767, 2015. http://eprint.iacr.org/2015/767.

[BLP04] Alex Biryukov, Joseph Lano, and Bart Preneel. Cryptanalysis of the alleged SecurID hash function. In Mitsuru Matsui and Robert J. Zuccherato, editors, *SAC 2003: 10th Annual International Workshop on Selected Areas in Cryptography*, volume 3006 of *Lecture Notes in Computer Science*, pages 130–144. Springer, Heidelberg, August 2004.

[BLR13] Alex Biryukov, Gaëtan Leurent, and Arnab Roy. Cryptanalysis of the "kindle" cipher. In Knudsen and Wu [KW13], pages 86–103.

[BMR+14] Andrey Bogdanov, Florian Mendel, Francesco Regazzoni, Vincent Rijmen, and Elmar Tischhauser. ALE: AES-based lightweight authenticated encryption. In Moriai [Mor14], pages 447–466.

[BN10] Alex Biryukov and Ivica Nikolic. Automatic search for related-key differential characteristics in byte-oriented block ciphers: Application to AES, Camellia, Khazad and others. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 322–344. Springer, Heidelberg, May 2010.

[BN14] Céline Blondeau and Kaisa Nyberg. Links between truncated differential and multidimensional linear properties of block ciphers and underlying attack complexities. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 165–182. Springer, Heidelberg, May 2014.

[BOK16] Achiya Bar-On and Nathan Keller. A $2^7 0$ attack on the full MISTY1. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 435–456. Springer, Heidelberg, August 2016.

[BP15] Alex Biryukov and Léo Perrin. On reverse-engineering S-boxes with hidden design criteria or structure. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 116–140. Springer, Heidelberg, August 2015.

[BPP+17] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A small present - towards reaching the limit of lightweight encryption. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems – CHES 2017*, volume 10529 of *Lecture Notes in Computer Science*, pages 321–345. Springer, Heidelberg, September 2017.

[BR00] Paulo Barreto and Vincent Rijmen. The Khazad legacy-level Block Cipher. First Open NESSIE Workshop, see also https://www.cosic.esat.kuleuven.be/nessie/workshop/submissions/khazad.zip, 2000.

[BR11] Andrey Bogdanov and Christian Rechberger. A 3-subset meet-in-the-middle attack: Cryptanalysis of the lightweight block cipher KTANTAN. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *SAC 2010: 17th Annual International Workshop on Selected Areas in Cryptography*, volume 6544 of *Lecture Notes in Computer Science*, pages 229–240. Springer, Heidelberg, August 2011.

[BR15] Elaine Barker and Allen Roginsky. NIST special publication 800-131a revision 1. *NIST Special Publication*, 800(131A):1–29, 2015.

[BŞ16] Adnan Baysal and Sühap Şahin. RoadRunneR: A small and fast bitslice block cipher for low cost 8-bit processors. In Tim Güneysu, Gregor Leander, and Amir Moradi, editors, *Lightweight Cryptography for Security and Privacy — LightSec 2015*, volume 9542 of *Lecture Notes in Computer Science*, pages 58–76, Berlin, Heidelberg, 2016. Springer International Publishing.

[BSK96] F.J. Bruwer, W. Smit, and G.J. Kuhn. Microchips and remote control devices comprising same, May 1996. US Patent 5,517,187.

[BSS+13] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK families of lightweight block ciphers. Cryptology ePrint Archive, Report 2013/404, 2013. http://eprint.iacr.org/2013/404.

[BSS+17] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. Notes on the design and analysis of SIMON and SPECK. Cryptology ePrint Archive, Report 2017/560, 2017. http://eprint.iacr.org/2017/560.

[BSW01] Alex Biryukov, Adi Shamir, and David Wagner. Real time cryptanalysis of A5/1 on a PC. In Bruce Schneier, editor, *Fast Software Encryption – FSE 2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 1–18. Springer, Heidelberg, April 2001.

[BTCS+15] Ray Beaulieu, Stefan Treatman-Clark, Douglas Shors, Bryan Weeks, Jason Smith, and Louis Wingers. The SIMON and SPECK lightweight block ciphers. In *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*, pages 1–6. IEEE, 2015.

[Bul13] Stanislav Bulygin. More on linear hulls of PRESENT-like ciphers and a cryptanalysis of full-round EPCBC-96. Cryptology ePrint Archive, Report 2013/028, 2013. http://eprint.iacr.org/2013/028.

[BVCZ08] Martin Boesgaard, Mette Vesterager, Thomas Christensen, and Erik Zenner. The stream cipher Rabbit. Available in the eSTREAM portfolio, a description is available at http://www.ecrypt.eu.org/stream/p3ciphers/rabbit/rabbit_p3.pdf, 2008.

[BW99] Alex Biryukov and David Wagner. Slide attacks. In Lars R. Knudsen, editor, *Fast Software Encryption – FSE'99*, volume 1636 of *Lecture Notes in Computer Science*, pages 245–259. Springer, Heidelberg, March 1999.

[Can05] D. Canright. A very compact S-box for AES. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2005*, volume 3659 of *Lecture Notes in Computer Science*, pages 441–455. Springer, Heidelberg, August / September 2005.

[Can06] Christophe De Cannière. Trivium: A stream cipher construction inspired by block cipher design principles. In Sokratis K. Katsikas, Javier Lopez, Michael Backes, Stefanos Gritzalis, and Bart Preneel, editors, *ISC 2006: 9th International Conference on Information Security*, volume 4176 of *Lecture Notes in Computer Science*, pages 171–186. Springer, Heidelberg, August / September 2006.

[CCHN15] Avik Chakraborti, Anupam Chattopadhyay, Muhammad Hassan, and Mridul Nandi. TriviA: A fast and secure authenticated encryption scheme. In Güneysu and Handschuh [GH15], pages 330–353.

[CDK09] Christophe De Cannière, Orr Dunkelman, and Miroslav Knežević. KATAN and KTANTAN - a family of small and efficient hardware-oriented block ciphers. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems – CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 272–288. Springer, Heidelberg, September 2009.

[CFG+15] Anne Canteaut, Thomas Fuhr, Henri Gilbert, María Naya-Plasencia, and Jean-René Reinhard. Multiple differential cryptanalysis of round-reduced PRINCE. In Cid and Rechberger [CR15], pages 591–610.

[CFG+17] Colin Chaigneau, Thomas Fuhr, Henri Gilbert, Jérémy Jean, and Jean-René Reinhard. Cryptanalysis of NORX v2.0. *IACR Transactions on Symmetric Cryptology*, 2017(1):156–174, 2017.

[CG00] Jean-Sébastien Coron and Louis Goubin. On Boolean and arithmetic masking against differential power analysis. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000*, volume 1965 of *Lecture Notes in Computer Science*, pages 231–237. Springer, Heidelberg, August 2000.

[CGTV15] Jean-Sébastien Coron, Johann Großschädl, Mehdi Tibouchi, and Praveen Kumar Vadnala. Conversion from arithmetic to Boolean masking with logarithmic complexity. In Leander [Lea15], pages 130–149.

[CLNP16] Anne Canteaut, Virginie Lallemand, and María Naya-Plasencia. Related-key attack on full-round PICARO. In Dunkelman and Keliher [DK16], pages 86–101.

[CMG+16] Stephen Checkoway, Jacob Maskiewicz, Christina Garman, Joshua Fried, Shaanan Cohney, Matthew Green, Nadia Heninger, Ralf-Philipp Weinmann, Eric Rescorla, and Hovav Shacham. A systematic analysis of the juniper dual EC incident. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 16: 23rd Conference on Computer and Communications Security*, pages 468–479. ACM Press, October 2016.

[CMM13] Mickaël Cazorla, Kevin Marquet, and Marine Minier. Survey and benchmark of lightweight block ciphers for wireless sensor networks. In Pierangela Samarati, editor, *SECRYPT 2013 - Proceedings of the 10th International Conference on Security and Cryptography, Reykjavík, Iceland, 29-31 July, 2013*, pages 543–548. SciTePress, 2013.

[CNO08] Nicolas T. Courtois, Karsten Nohl, and Sean O'Neil. Algebraic attacks on the crypto-1 stream cipher in mifare classic and oyster cards. Cryptology ePrint Archive, Report 2008/166, 2008. http://eprint.iacr.org/2008/166.

[Cop94] Don Coppersmith. The Data Encryption Standard (DES) and its strength against attacks. *IBM journal of research and development*, 38(3):243–250, 1994.

[CR15] Carlos Cid and Christian Rechberger, editors. *Fast Software Encryption – FSE 2014*, volume 8540 of *Lecture Notes in Computer Science*. Springer, Heidelberg, March 2015.

[CRY17] CRYPTREC Lightweight Cryptography Working Group. CRYPTREC cryptographic technology guideline (lightweight cryptography). Available online at http://www.cryptrec.go.jp/report/cryptrec-gl-0001-2016-e.pdf, March 2017.

[CT16] Jung Hee Cheon and Tsuyoshi Takagi, editors. *Advances in Cryptology – ASIACRYPT 2016, Part I*, volume 10031 of *Lecture Notes in Computer Science*. Springer, Heidelberg, December 2016.

[CWP12] Jiazhe Chen, Meiqin Wang, and Bart Preneel. Impossible differential cryptanalysis of the lightweight block ciphers TEA, XTEA and HIGHT. In Mitrokotsa and Vaudenay [MV12b], pages 117–137.

[CY04] Scott Contini and Yiqun Lisa Yin. Fast software-based attacks on SecurID. In Roy and Meier [RM04], pages 454–471.

[CYK+12] Jiali Choy, Huihui Yap, Khoongming Khoo, Jian Guo, Thomas Peyrin, Axel Poschmann, and Chik How Tan. SPN-hash: Improving the provable resistance against differential collision attacks. In Mitrokotsa and Vaudenay [MV12b], pages 270–286.

[DBG+15] Dumitru-Daniel Dinu, Alex Biryukov, Johann Großschädl, Dmitry Khovratovich, Yann Le Corre, and Léo Perrin. FELICS – Fair Evaluation of Lightweight Cryptographic Systems. In *NIST Workshop on Lightweight Cryptography 2015*. National Institute of Standards and Technology (NIST), 2015.

[DDKS16] Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. Key recovery attacks on iterated Even-Mansour encryption schemes. *Journal of Cryptology*, 29(4):697–728, October 2016.

[DEKM16] Christoph Dobraunig, Maria Eichlseder, Daniel Kales, and Florian Mendel. 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.

[DEMS16] C. Dobraunig, M. Eichlseder, F. Mendel, and M Schläffer. Ascon v1.2. Candidate for the CAESAR Competition. See also http://ascon.iaik.tugraz.at/, 2016.

[DF16] Patrick Derbez and Pierre-Alain Fouque. Automatic search of meet-in-the-middle and impossible differential attacks. In Robshaw and Katz [RK16], pages 157–184.

[DFJ13] Patrick Derbez, Pierre-Alain Fouque, and Jérémy Jean. Improved key recovery attacks on reduced-round AES in the single-key setting. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 371–387. Springer, Heidelberg, May 2013.

[DGV94] Joan Daemen, René Govaerts, and Joos Vandewalle. A new approach to block cipher design. In Ross J. Anderson, editor, *Fast Software Encryption – FSE'93*, volume 809 of *Lecture Notes in Computer Science*, pages 18–32. Springer, Heidelberg, December 1994.

[DH17] Elena Dubrova and Martin Hell. Espresso: A stream cipher for 5G wireless communication systems. *Cryptography and Communications*, 9(2):273–289, Mar 2017.

[DHW+12] B. Driessen, R. Hund, C. Willems, C. Paar, and T. Holz. Don't trust satellite phones: A security analysis of two satphone standards. In *2012 IEEE Symposium on Security and Privacy*, pages 128–142, May 2012.

[DJ15] Itai Dinur and Jérémy Jean. Cryptanalysis of FIDES. In Cid and Rechberger [CR15], pages 224–240.

[DK16] Orr Dunkelman and Liam Keliher, editors. *SAC 2015: 22nd Annual International Workshop on Selected Areas in Cryptography*, volume 9566 of *Lecture Notes in Computer Science*. Springer, Heidelberg, August 2016.

[Dol10] V. Dolmatov. Gost 28147-89: Encryption, decryption, and message authentication code (mac) algorithms. http://www.rfc-editor.org/rfc/rfc5830.txt, March 2010. RFC 5830.

[DP15] Patrick Derbez and Léo Perrin. Meet-in-the-middle attacks and structural analysis of round-reduced PRINCE. In Leander [Lea15], pages 190–216.

[DPU+16] Daniel Dinu, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, Johann Großschädl, and Alex Biryukov. Design strategies for ARX with provable bounds: Sparx and LAX. In Cheon and Takagi [CT16], pages 484–513.

[DPVAR00] Joan Daemen, Michaël Peeters, Gilles Van Assche, and Vincent Rijmen. Nessie proposal: NOEKEON. First Open NESSIE Workshop, see also http://gva.noekeon.org/papers/2000-NESSIE-Noekeon-Spec.pdf, 2000.

[DR98] Joan Daemen and Vincent Rijmen. AES proposal: Rijndael. AES submission. See also http://csrc.nist.gov/archive/aes/rijndael/, 1998.

[DR00] Joan Daemen and Vincent Rijmen. The block cipher BKSQ. In Jean-Jacques Quisquater and Bruce Schneier, editors, *Smart Card Research and Applications: Third International Conference, CARDIS'98, Louvain-la-Neuve, Belgium, September 14-16, 1998. Proceedings*, volume 1820 of *Lecture Notes in Computer Science*, pages 236–245, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[Dri02] Kevin Driscoll. BeepBeep: Embedded real-time encryption. In Joan Daemen and Vincent Rijmen, editors, *Fast Software Encryption – FSE 2002*, volume 2365 of *Lecture Notes in Computer Science*, pages 164–178. Springer, Heidelberg, February 2002.

[DRL11] M. David, D. C. Ranasinghe, and T. Larsen. A2U2: A stream cipher for printed electronics RFID tags. In *2011 IEEE International Conference on RFID*, pages 176–183, April 2011.

[DS11] Itai Dinur and Adi Shamir. Breaking Grain-128 with dynamic cube attacks. In Joux [Jou11], pages 167–187.

[EJ03] Patrik Ekdahl and Thomas Johansson. A new version of the stream cipher SNOW. In Kaisa Nyberg and Howard M. Heys, editors, *SAC 2002: 9th Annual International Workshop on Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 47–61. Springer, Heidelberg, August 2003.

[EKM+08] Thomas Eisenbarth, Timo Kasper, Amir Moradi, Christof Paar, Mahmoud Salmasizadeh, and Mohammad T. Manzuri Shalmani. On the power of power analysis in the real world: A complete break of the keeloqcode hopping scheme. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 203–220. Springer, Heidelberg, August 2008.

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

[ESSS12] Daniel Engels, Markku-Juhani O. Saarinen, Peter Schweitzer, and Eric M. Smith. The Hummingbird-2 lightweight authenticated encryption algorithm. In Ari Juels and Christof Paar, editors, *RFID. Security and Privacy: 7th International Workshop, RFIDSec 2011, Amherst, USA, June 26-28, 2011, Revised Selected Papers*, volume 7055 of *Lecture Notes in Computer Science*, pages 19–31, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[ETS06] ETSI/SAGE. Specification of the 3GPP confidentiality and integrity algorithms UEA2 & UIA2. Document 2: SNOW 3G specification. Technical report, ETSI/Sage, September 2006. Available at http://www.gsma.com/aboutus/wp-content/uploads/2014/12/snow3gspec.doc (Microsoft Word document).

[ETS11] ETSI/Sage. Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 4 : Design and Evaluation Report. Technical report, ETSI/Sage, September 2011. Available at http://www.gsma.com/aboutus/wp-content/uploads/2014/12/EEA3_EIA3_Design_Evaluation_v2_0.pdf.

[ETS14] ETSI/Sage. Universal Mobile Telecommunications System (UMTS); LTE; Specification of the TUAK algorithm set: A second example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 1: Algorithm specification (3GPP TS 35.231 version 12.1.0 Release 12). Technical report, ETSI/Sage, October 2014. Available at http://www.etsi.org/deliver/etsi_ts/135200_135299/135231/12.01.00_60/ts_135231v120100p.pdf.

[Fed15] Federal Agency on Technical Regulation and Metrology (GOST). (GOST R 34.12–2015) information technology – cryptographic data security – block ciphers, 2015. http://tc26.ru/en/standard/gost/GOST_R_34_12_2015_ENG.pdf.

[FL01] Scott R. Fluhrer and Stefan Lucks. Analysis of the E0 encryption system. In Vaudenay and Youssef [VY01], pages 38–48.

[FM15] Thomas Fuhr and Brice Minaud. Match box meet-in-the-middle attack against KATAN. In Cid and Rechberger [CR15], pages 61–81.

[FMS01] Scott R. Fluhrer, Itsik Mantin, and Adi Shamir. Weaknesses in the key scheduling algorithm of RC4. In Vaudenay and Youssef [VY01], pages 1–24.

[FWR05] Martin Feldhofer, Johannes Wolkerstorfer, and Vincent Rijmen. AES implementation on a grain of sand. *IEE Proceedings-Information Security*, 152(1):13–20, 2005.

[FWS+03] Niels Ferguson, Doug Whiting, Bruce Schneier, John Kelsey, Stefan Lucks, and Tadayoshi Kohno. Helix: Fast encryption and authentication in a single cryptographic primitive. In Thomas Johansson, editor, *Fast Software Encryption – FSE 2003*, volume 2887 of *Lecture Notes in Computer Science*, pages 330–346. Springer, Heidelberg, February 2003.

[FZ14] Xiutao Feng and Fan Zhang. Cryptanalysis on the authenticated cipher sablier. In Man Ho Au, Barbara Carminati, and C.-C. Jay Kuo, editors, *Network and System Security: 8th International Conference, NSS 2014, Xi'an, China, October 15-17, 2014, Proceedings*, volume 8792 of *Lecture Notes in Computer Science*, pages 198–208, Cham, 2014. Springer International Publishing.

[GdKGV14] Flavio D. Garcia, Gerhard de Koning Gans, and Roel Verdult. Wirelessly lockpicking a smart card reader. *International Journal of Information Security*, 13(5):403–420, 2014.

[GGNS13] Benoît Gérard, Vincent Grosso, María Naya-Plasencia, and François-Xavier Standaert. Block ciphers that are easier to mask: How far can we go? In Bertoni and Coron [BC13], pages 383–399.

[GH15] Tim Güneysu and Helena Handschuh, editors. *Cryptographic Hardware and Embedded Systems – CHES 2015*, volume 9293 of *Lecture Notes in Computer Science*. Springer, Heidelberg, September 2015.

[GKA+10] Kris Gaj, Jens-Peter Kaps, Venkata Amirineni, Marcin Rogawski, Ekawat Homsirikamol, and Benjamin Y Brewster. ATHENa-automated tool for hardware evaluation: Toward fair and comprehensive benchmarking of cryptographic hardware using FPGAs. In *2010 International Conference on Field Programmable Logic and Applications (FPL)*, pages 414–421. IEEE, 2010.

[GLS+14] Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, Kerem Varici, François Durvaux Anthony Journault, Lubos Gaspar, and Stéphanie Kerckhof. SCREAM & iSCREAM Side-Channel Resistant Authenticated Encryption with Masking. Candidate for the CAESAR Competition. See also http://perso.uclouvain.be/fstandae/SCREAM/, 2014.

[GLSV15] Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, and Kerem Varici. LS-designs: Bitslice encryption for efficient masked software implementations. In Cid and Rechberger [CR15], pages 18–37.

[GNL11] Zheng Gong, Svetla Nikova, and Yee Wei Law. KLEIN: A new family of lightweight block ciphers. In Ari Juels and Christof Paar, editors, *RFID. Security and Privacy - 7th International Workshop, RFIDSec*, volume 7055 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2011.

[Gol97] Jovan Dj. Golic. Cryptanalysis of alleged A5 stream cipher. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 239–255. Springer, Heidelberg, May 1997.

[Gol13] Jovan Dj. Golic. Cryptanalytic attacks on MIFARE classic protocol. In Ed Dawson, editor, *Topics in Cryptology – CT-RSA 2013*, volume 7779 of *Lecture Notes in Computer Science*, pages 239–258. Springer, Heidelberg, February / March 2013.

[GP99] Louis Goubin and Jacques Patarin. DES and differential power analysis (the "duplication" method). In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES'99*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer, Heidelberg, August 1999.

[GPP11] Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON family of lightweight hash functions. In Rogaway [Rog11], pages 222–239.

[GPPR11] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In Preneel and Takagi [PT11], pages 326–341.

[GvRVWS10] Flavio D. Garcia, Peter van Rossum, Roel Verdult, and Ronny Wichers Schreur. Dismantling SecureMemory, CryptoMemory and CryptoRF. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 250–259, New York, NY, USA, 2010. ACM.

[Hal09] Shai Halevi, editor. *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*. Springer, Heidelberg, August 2009.

[HI10] Seokhie Hong and Tetsu Iwata, editors. *Fast Software Encryption – FSE 2010*, volume 6147 of *Lecture Notes in Computer Science*. Springer, Heidelberg, February 2010.

[HIK+11] Shoichi Hirose, Kota Ideguchi, Hidenori Kuwakado, Toru Owada, Bart Preneel, and Hirotaka Yoshida. A lightweight 256-bit hash function for hardware and low-end devices: Lesamnta-LW. In Rhee and Nyang [RN11], pages 151–168.

[HJ11] Martin Hell and Thomas Johansson. Breaking the stream ciphers F-FCSR-H and F-FCSR-16 in real time. *Journal of Cryptology*, 24(3):427–445, July 2011.

[HJM07] Martin Hell, Thomas Johansson, and Willi Meier. Grain: a stream cipher for constrained environments. *Int. J. Wire. Mob. Comput.*, 2(1):86–93, May 2007.

[HKM17] Matthias Hamann, Matthias Krause, and Willi Meier. LIZARD – A lightweight stream cipher for power-constrained devices. *IACR Transactions on Symmetric Cryptology*, 2017(1):45–79, 2017.

[HLK+14] Deukjo Hong, Jung-Keun Lee, Dong-Chan Kim, Daesung Kwon, Kwon Ho Ryu, and Dong-Geon Lee. LEA: A 128-bit block cipher for fast encryption on common processors. In Yongdae Kim, Heejo Lee, and Adrian Perrig, editors, *WISA 13: 14th International Workshop on Information Security Applications*, volume 8267 of *Lecture Notes in Computer Science*, pages 3–27. Springer, Heidelberg, August 2014.

[HSH+06] Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bon-Seok Koo, Changhoon Lee, Donghoon Chang, Jesang Lee, Kitae Jeong, Hyun Kim, Jongsung Kim, and Seongtaek Chee. HIGHT: A new block cipher suitable for low-resource device. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems – CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 46–59. Springer, Heidelberg, October 2006.

[IKD+08] Sebastiaan Indesteege, Nathan Keller, Orr Dunkelman, Eli Biham, and Bart Preneel. A practical attack on KeeLoq. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 1–18. Springer, Heidelberg, April 2008.

[Iso13] Takanori Isobe. A single-key attack on the full GOST block cipher. *Journal of Cryptology*, 26(1):172–189, January 2013.

[ISSK09] Maryam Izadi, Babak Sadeghiyan, Seyed Saeed Sadeghian, and Hossein Arabnezhad Khanooki. MIBS: A new lightweight block cipher. In Juan A. Garay, Atsuko Miyaji, and Akira Otsuka, editors, *CANS 09: 8th International Conference on Cryptology and Network Security*, volume 5888 of *Lecture Notes in Computer Science*, pages 334–348. Springer, Heidelberg, December 2009.

[Iwa06] Tetsu Iwata. New blockcipher modes of operation with beyond the birthday bound security. In Robshaw [Rob06], pages 310–327.

[JK12] Goce Jakimoski and Samant Khajuria. ASC-1: An authenticated encryption stream cipher. In Miri and Vaudenay [MV12a], pages 356–372.

[JNP14a] Jérémy Jean, Ivica Nikolìc, and Thomas Peyrin. Joltik v1. Candidate for the CAESAR Competition. See also http://www1.spms.ntu.edu.sg/~syllab/m/index.php/Joltik, 2014.

[JNP14b] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and keys for block ciphers: The TWEAKEY framework. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 274–288. Springer, Heidelberg, December 2014.

[Jou11] Antoine Joux, editor. *Fast Software Encryption – FSE 2011*, volume 6733 of *Lecture Notes in Computer Science*. Springer, Heidelberg, February 2011.

[JPS17] Jérémy Jean, Thomas Peyrin, and Siang Meng Sim. Optimizing implementations of lightweight building blocks. Cryptology ePrint Archive, Report 2017/101, 2017. http://eprint.iacr.org/2017/101.

[JSV17] Anthony Journault, François-Xavier Standaert, and Kerem Varici. Improving the security and efficiency of block ciphers based on ls-designs. *Designs, Codes and Cryptography*, 82(1):495–509, 2017.

[KDH13] Ferhat Karakoç, Hüseyin Demirci, and A. Emre Harmancı. ITUbee: A software oriented lightweight block cipher. In Gildas Avoine and Orhun Kara, editors, *Lightweight Cryptography for Security and Privacy: LightSec 2013*, volume 8162 of *Lecture Notes in Computer Science*, pages 16–27, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[Ker83] Auguste Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX:5–38, January 1883. Available online on the website of Fabien Petitcolas: http://www.petitcolas.net/kerckhoffs/crypto_militaire_1.pdf (the article was written in French).

[KG16] Pierre Karpman and Benjamin Grégoire. The LITTLUN S-box and the FLY block cipher. In *Lightweight Cryptography Workshop 2016, October 17-18 (informal proceedings)*. National Institute of Standards and Technology, 2016.

[KHK11] Bonwook Koo, Deukjo Hong, and Daesung Kwon. Related-key attack on the full HIGHT. In Rhee and Nyang [RN11], pages 49–67.

[KLPR10] Lars R. Knudsen, Gregor Leander, Axel Poschmann, and Matthew J. B. Robshaw. PRINTcipher: A block cipher for IC-printing. In Mangard and Standaert [MS10], pages 16–32.

[KR14] Dmitry Khovratovich and Christian Rechberger. The LOCAL attack: Cryptanalysis of the authenticated encryption scheme ALE. In Tanja Lange, Kristin Lauter, and Petr Lisonek, editors, *SAC 2013: 20th Annual International Workshop on Selected Areas in Cryptography*, volume 8282 of *Lecture Notes in Computer Science*, pages 174–184. Springer, Heidelberg, August 2014.

[KRW99] Lars R. Knudsen, Matthew J. B. Robshaw, and David Wagner. Truncated differentials and Skipjack. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 165–180. Springer, Heidelberg, August 1999.

[KSW97] John Kelsey, Bruce Schneier, and David Wagner. Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In Yongfei Han, Tatsuaki Okamoto, and Sihan Qing, editors, *ICICS 97: 1st International Conference on Information and Communication Security*, volume 1334 of *Lecture Notes in Computer Science*, pages 233–246. Springer, Heidelberg, November 1997.

[KW01] Lars Knudsen and David Wagner. On the structure of Skipjack. *Discrete Applied Mathematics*, 111(1–2):103 – 116, 2001. Coding and Cryptology.

[KW13] Lars R. Knudsen and Huapeng Wu, editors. *SAC 2012: 19th Annual International Workshop on Selected Areas in Cryptography*, volume 7707 of *Lecture Notes in Computer Science*. Springer, Heidelberg, August 2013.

[LAAZ11] Gregor Leander, Mohamed Ahmed Abdelraheem, Hoda AlKhzaimi, and Erik Zenner. A cryptanalysis of PRINTcipher: The invariant subspace attack. In Rogaway [Rog11], pages 206–221.

[Lea14] Gregor Leander. Lightweight block cipher design. Presentation slides available at https://summerschool-croatia.cs.ru.nl/2014/slides/Lightweight%20Block%20Cipher%20Design.pdf, 2014.

[Lea15] Gregor Leander, editor. *Fast Software Encryption – FSE 2015*, volume 9054 of *Lecture Notes in Computer Science*. Springer, Heidelberg, March 2015.

[Leu16a] Gaëtan Leurent. Differential forgery attack against LAC. In Dunkelman and Keliher [DK16], pages 217–224.

[Leu16b] Gaëtan Leurent. Improved differential-linear cryptanalysis of 7-round chaskey with partitioning. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 344–371. Springer, Heidelberg, May 2016.

[LGS17] Guozhen Liu, Mohona Ghosh, and Ling Song. Security analysis of SKINNY under related-tweakey settings (long paper). *IACR Transactions on Symmetric Cryptology*, 2017(3):37–72, 2017.

[LK06] Chae Hoon Lim and Tymur Korkishko. mCrypton - a lightweight block cipher for security of low-cost RFID tags and sensors. In Jooseok Song, Taekyoung Kwon, and Moti Yung, editors, *WISA 05: 6th International Workshop on Information Security Applications*, volume 3786 of *Lecture Notes in Computer Science*, pages 243–258. Springer, Heidelberg, August 2006.

[LLLS14] Ruilin Li, Heng Li, Chao Li, and Bing Sun. A low data complexity attack on the GMR-2 cipher used in the satellite phones. In Moriai [Mor14], pages 485–501.

[LM91] Xuejia Lai and James L. Massey. A proposal for a new block encryption standard. In Ivan Damgård, editor, *Advances in Cryptology – EUROCRYPT'90*, volume 473 of *Lecture Notes in Computer Science*, pages 389–404. Springer, Heidelberg, May 1991.

[LMR15] Gregor Leander, Brice Minaud, and Sondre Rønjom. A generic approach to invariant subspace attacks: Cryptanalysis of robin, iSCREAM and Zorro. In Oswald and Fischlin [OF15], pages 254–283.

[LMV05] Yi Lu, Willi Meier, and Serge Vaudenay. The conditional correlation attack: A practical attack on bluetooth encryption. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 97–117. Springer, Heidelberg, August 2005.

[LN15] Virginie Lallemand and María Naya-Plasencia. Cryptanalysis of KLEIN. In Cid and Rechberger [CR15], pages 451–470.

[LPPS07] Gregor Leander, Christof Paar, Axel Poschmann, and Kai Schramm. New lightweight DES variants. In Biryukov [Bir07], pages 196–210.

[LR17] Virginie Lallemand and Shahram Rasoolzadeh. Differential cryptanalysis of 18-round PRIDE. In Arpita Patra and Nigel P. Smart, editors, *Progress in Cryptology – INDOCRYPT 2017*, volume 10698 of *Lecture Notes in Computer Science*, page To appear, Cham, 2017. Springer International Publishing.

[LRW02] Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 31–46. Springer, Heidelberg, August 2002.

[LST+09] Stefan Lucks, Andreas Schuler, Erik Tews, Ralf-Philipp Weinmann, and Matthias Wenzel. Attacks on the DECT authentication mechanisms. In Marc Fischlin, editor, *Topics in Cryptology – CT-RSA 2009*, volume 5473 of *Lecture Notes in Computer Science*, pages 48–65. Springer, Heidelberg, April 2009.

[LT11] Javier Lopez and Gene Tsudik, editors. *ACNS 11: 9th International Conference on Applied Cryptography and Network Security*, volume 6715 of *Lecture Notes in Computer Science*. Springer, Heidelberg, June 2011.

[Lu09] Jiqiang Lu. Related-key rectangle attack on 36 rounds of the xtea block cipher. *International Journal of Information Security*, 8(1):1–11, Feb 2009.

[LV04] Yi Lu and Serge Vaudenay. Faster correlation attack on Bluetooth keystream generator E0. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 407–425. Springer, Heidelberg, August 2004.

[MAM17] Vasily Mikhalev, Frederik Armknecht, and Christian Müller. On ciphers that continuously access the non-volatile key. *IACR Transactions on Symmetric Cryptology*, 2016(2):52–79, 2017.

[Mat94] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In Tor Helleseth, editor, *Advances in Cryptology – EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, Heidelberg, May 1994.

[Mat97] Mitsuru Matsui. New block encryption algorithm MISTY. In Biham [Bih97], pages 54–68.

[MBTM17] Kerry A. McKay, Larry Bassham, Meltem Sönmez Turan, and Nicky Mouha. NISTIR 8114 – report on lightweight cryptography. Available on the NIST website: http://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8114.pdf, 2017.

[MDS11] Hamid Mala, Mohammad Dakhilalian, and Mohsen Shakiba. Impossible differential attacks on 13-round clefia-128. *Journal of Computer Science and Technology*, 26(4):744–750, Jul 2011.

[Men15] Bart Mennink. Optimally secure tweakable blockciphers. In Leander [Lea15], pages 428–448.

[Min13] Marine Minier. On the security of piccolo lightweight block cipher against related-key impossible differentials. In Goutam Paul and Serge Vaudenay, editors, *Progress in Cryptology - INDOCRYPT 2013: 14th International Conference in Cryptology in India*, volume 8250 of *Lecture Notes in Computer Science*, pages 308–318. Springer, Heidelberg, December 2013.

[MM15] Mitsuru Matsui and Yumiko Murakami. Aes smaller than s-box. In Thomas Eisenbarth and Erdinç Öztürk, editors, *Lightweight Cryptography for Security and Privacy: Third International Workshop, LightSec 2014, Istanbul, Turkey, September 1-2, 2014, Revised Selected Papers*, volume 8898 of *Lecture Notes in Computer Science*, pages 51–66, Cham, 2015. Springer International Publishing.

[MMH+14] Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel, and Ingrid Verbauwhede. Chaskey: An efficient MAC algorithm for 32-bit microcontrollers. In Antoine Joux and Amr M. Youssef, editors, *SAC 2014: 21st Annual International Workshop on Selected Areas in Cryptography*, volume 8781 of *Lecture Notes in Computer Science*, pages 306–323. Springer, Heidelberg, August 2014.

[Mor14] Shiho Moriai, editor. *Fast Software Encryption – FSE 2013*, volume 8424 of *Lecture Notes in Computer Science*. Springer, Heidelberg, March 2014.

[Mou15] Nicky Mouha. Chaskey: a MAC algorithm for microcontrollers – status update and proposal of Chaskey-12 –. Cryptology ePrint Archive, Report 2015/1182, 2015. http://eprint.iacr.org/2015/1182.

[MPL+11] Amir Moradi, Axel Poschmann, San Ling, Christof Paar, and Huaxiong Wang. Pushing the limits: A very compact and a threshold implementation of AES. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 69–88. Springer, Heidelberg, May 2011.

[MS10] Stefan Mangard and François-Xavier Standaert, editors. *Cryptographic Hardware and Embedded Systems – CHES 2010*, volume 6225 of *Lecture Notes in Computer Science*. Springer, Heidelberg, August 2010.

[Mul04] Frédéric Muller. Differential attacks against the Helix stream cipher. In Roy and Meier [RM04], pages 94–108.

[MV12a] Ali Miri and Serge Vaudenay, editors. *SAC 2011: 18th Annual International Workshop on Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*. Springer, Heidelberg, August 2012.

[MV12b] Aikaterini Mitrokotsa and Serge Vaudenay, editors. *AFRICACRYPT 12: 5th International Conference on Cryptology in Africa*, volume 7374 of *Lecture Notes in Computer Science*. Springer, Heidelberg, July 2012.

[NESP08] Karsten Nohl, David Evans, Starbug Starbug, and Henryk Plötz. Reverse-engineering a cryptographic RFID tag. In *USENIX security symposium*, volume 28, 2008.

[NTW10] Karsten Nohl, Erik Tews, and Ralf-Philipp Weinmann. Cryptanalysis of the DECT standard cipher. In Hong and Iwata [HI10], pages 1–18.

[NW97] R. M. Needham and D. J. Wheeler. Tea extensions. Technical report, Cambridge University, Cambridge, UK, October 1997.

[OBSC10] Dag Arne Osvik, Joppe W. Bos, Deian Stefan, and David Canright. Fast software AES encryption. In Hong and Iwata [HI10], pages 75–93.

[OC16] Colin O'Flynn and Zhizhang Chen. Power analysis attacks against IEEE 802.15.4 nodes. In *Constructive Side-Channel Analysis and Secure Design – COSADE 2016*, volume 9689 of *Lecture Notes in Computer Science*, Cham, 2016. Springer International Publishing.

[OF15] Elisabeth Oswald and Marc Fischlin, editors. *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*. Springer, Heidelberg, April 2015.

[Per17] Léo Perrin. More reverse-engineered S-boxes. Presentation at the rump session of ESC'2017. Slides available at https://www.cryptolux.org/mediawiki-esc2017/images/2/2e/Rump.pdf, 2017.

[PK15] Léo Perrin and Dmitry Khovratovich. Collision spectrum, entropy loss, T-sponges, and cryptanalysis of GLUON-64. In Cid and Rechberger [CR15], pages 82–103.

[PLW10] Axel Poschmann, San Ling, and Huaxiong Wang. 256 bit standardized crypto for 650 GE - GOST revisited. In Mangard and Standaert [MS10], pages 219–233.

[PMA07] Lea Troels Møller Pedersen, Carsten Valdemar Munk, and Lisbet Møller Andersen. Cryptography – the rise and fall of DVD encryption. Available online at http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=3672D97255B2446765DA47DA97960CDF?doi=10.1.1.118.6103&rep=rep1&type=pdf, 2007.

[PRC12] Gilles Piret, Thomas Roche, and Claude Carlet. PICARO - a block cipher allowing efficient higher-order side-channel resistance. In Feng Bao, Pierangela Samarati, and Jianying Zhou, editors, *ACNS 12: 10th International Conference on Applied Cryptography and Network Security*, volume 7341 of *Lecture Notes in Computer Science*, pages 311–328. Springer, Heidelberg, June 2012.

[Pre95] Bart Preneel, editor. *Fast Software Encryption – FSE'94*, volume 1008 of *Lecture Notes in Computer Science*. Springer, Heidelberg, December 1995.

[PT11] Bart Preneel and Tsuyoshi Takagi, editors. *Cryptographic Hardware and Embedded Systems – CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*. Springer, Heidelberg, September / October 2011.

[QHS17] Kexin Qiao, Lei Hu, and Siwei Sun. Differential analysis on simeck and simon with dynamic key-guessing techniques. In Olivier Camp, Steven Furnell, and Paolo Mori, editors, *Information Systems Security and Privacy: Second International Conference, ICISSP 2016, Rome, Italy, February 19-21, 2016, Revised Selected Papers*, volume 691 of *Communications in Computer and Information Science*, pages 64–85, Cham, 2017. Springer International Publishing.

[Riv95] Ronald L. Rivest. The RC5 encryption algorithm. In Preneel [Pre95], pages 86–96.

[RK16] Matthew Robshaw and Jonathan Katz, editors. *Advances in Cryptology – CRYPTO 2016, Part II*, volume 9815 of *Lecture Notes in Computer Science*. Springer, Heidelberg, August 2016.

[RM04] Bimal K. Roy and Willi Meier, editors. *Fast Software Encryption – FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*. Springer, Heidelberg, February 2004.

[RN11] Kyung Hyune Rhee and DaeHun Nyang, editors. *ICISC 10: 13th International Conference on Information Security and Cryptology*, volume 6829 of *Lecture Notes in Computer Science*. Springer, Heidelberg, December 2011.

[Rob06] Matthew J. B. Robshaw, editor. *Fast Software Encryption – FSE 2006*, volume 4047 of *Lecture Notes in Computer Science*. Springer, Heidelberg, March 2006.

[Rog11] Phillip Rogaway, editor. *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*. Springer, Heidelberg, August 2011.

[ROSW16] Eyal Ronen, Colin O'Flynn, Adi Shamir, and Achi-Or Weingarten. IoT goes nuclear: Creating a Zigbee chain reaction. Cryptology ePrint Archive, Report 2016/1047, 2016. http://eprint.iacr.org/2016/1047.

[SA12] Yu Sasaki and Kazumaro Aoki. Improved integral analysis on tweaked lesamnta. In Howon Kim, editor, *ICISC 11: 14th International Conference on Information Security and Cryptology*, volume 7259 of *Lecture Notes in Computer Science*, pages 1–17. Springer, Heidelberg, November / December 2012.

[Saa11] Markku-Juhani O. Saarinen. Cryptanalysis of Hummingbird-1. In Joux [Jou11], pages 328–341.

[Saa14] Markku-Juhani O. Saarinen. Related-key attacks against full Hummingbird-2. In Moriai [Mor14], pages 467–482.

[SHY16] Ling Song, Zhangjie Huang, and Qianqian Yang. Automatic differential analysis of ARX block ciphers with application to SPECK and LEA. In Joseph K. Liu and Ron Steinfeld, editors, *ACISP 16: 21st Australasian Conference on Information Security and Privacy, Part II*, volume 9723 of *Lecture Notes in Computer Science*, pages 379–394. Springer, Heidelberg, July 2016.

[SIH+11] Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. Piccolo: An ultra-lightweight blockcipher. In Preneel and Takagi [PT11], pages 342–357.

[SM10] Tomoyasu Suzaki and Kazuhiko Minematsu. Improving the generalized Feistel. In Hong and Iwata [HI10], pages 19–39.

[SMMK13] Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. TWINE : A lightweight block cipher for multiple platforms. In Knudsen and Wu [KW13], pages 339–354.

[SMVP11] Gautham Sekar, Nicky Mouha, Vesselin Velichkov, and Bart Preneel. Meet-in-the-middle attacks on reduced-round XTEA. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 250–267. Springer, Heidelberg, February 2011.

[Sol15] H. Soleimany. Self-similarity cryptanalysis of the block cipher itubee. *IET Information Security*, 9(3):179–184, 2015.

[SPGQ06] François-Xavier Standaert, Gilles Piret, Neil Gershenfeld, and Jean-Jacques Quisquater. Sea: A scalable encryption algorithm for small embedded applications. In Josep Domingo-Ferrer, Joachim Posegga, and Daniel Schreckling, editors, *Smart Card Research and Advanced Applications: CARDIS 2006*, volume 3928 of *Lecture Notes in Computer Science*, pages 222–236, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[SPR+04] François-Xavier Standaert, Gilles Piret, Gaël Rouvroy, Jean-Jacques Quisquater, and Jean-Didier Legat. ICEBERG: An involutional cipher efficient for block encryption in reconfigurable hardware. In Roy and Meier [RM04], pages 279–299.

[SS16] Peter Schwabe and Ko Stoffelen. All the AES you need on Cortex-M3 and M4. In Roberto Avanzi and Howard M. Heys, editors, *SAC 2016: 23rd Annual International Workshop on Selected Areas in Cryptography*, volume 10532 of *Lecture Notes in Computer Science*, pages 180–194. Springer, Heidelberg, August 2016.

[SSA+07] Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai, and Tetsu Iwata. The 128-bit blockcipher CLEFIA (extended abstract). In Biryukov [Bir07], pages 181–195.

[ST17] Yu Sasaki and Yosuke Todo. New differential bounds and division property of lilliput: Block cipher with extended generalized feistel network. In Roberto Avanzi and Howard Heys, editors, *Selected Areas in Cryptography – SAC 2016: 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers*, volume 10532 of *Lecture Notes in Computer Science*, pages 264–283, Cham, 2017. Springer International Publishing.

[SWJS12] Yue Sun, Meiqin Wang, Shujia Jiang, and Qiumei Sun. Differential cryptanalysis of reduced-round ICEBERG. In Mitrokotsa and Vaudenay [MV12b], pages 155–171.

[TLS16] Yosuke Todo, Gregor Leander, and Yu Sasaki. Nonlinear invariant attack - practical attack on full SCREAM, iSCREAM, and Midori64. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 3–33. Springer, Heidelberg, December 2016.

[Tod17] Yosuke Todo. Integral cryptanalysis on full MISTY1. *Journal of Cryptology*, 30(3):920–959, July 2017.

[UMHA16] Rei Ueno, Sumio Morioka, Naofumi Homma, and Takafumi Aoki. A high throughput/gate AES hardware architecture by compressing encryption and decryption datapaths - toward efficient CBC-mode implementation. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems – CHES 2016*, volume 9813 of *Lecture Notes in Computer Science*, pages 538–558. Springer, Heidelberg, August 2016.

[U.S98] U.S. Department Of Commerce/National Institute of Standards and Technology. Skipjack and KEA algorithms specifications, v2.0, 1998. http://csrc.nist.gov/groups/ST/toolkit/documents/skipjack/skipjack.pdf.

[U.S99] U.S. Department Of Commerce/National Institute of Standards and Technology. Data Encryption Standard. *Federal Information Processing Standards Publication (FIPS)*, 1999. Available on the NIST website: http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf.

[U.S15] U.S. Department Of Commerce/National Institute of Standards and Technology. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. *Federal Information Processing Standards Publication (FIPS)*, 2015. Available on the NIST website: http://ws680.nist.gov/publication/get_pdf.cfm?pub_id=919061.

[VGB12] Roel Verdult, Flavio D. Garcia, and Josep Balasch. Gone in 360 seconds: Hijacking with hitag2. In *Proceedings of the 21st USENIX Conference on Security Symposium*, Security'12, pages 37–37, Berkeley, CA, USA, 2012. USENIX Association.

[VGE13] Roel Verdult, Flavio D Garcia, and Baris Ege. Dismantling Megamos crypto: Wirelessly lockpicking a vehicle immobilizer. In *Supplement to the 22nd USENIX Security Symposium (USENIX Security 13)*, pages 703–718. USENIX Association, August 2013.

[VY01] Serge Vaudenay and Amr M. Youssef, editors. *SAC 2001: 8th Annual International Workshop on Selected Areas in Cryptography*, volume 2259 of *Lecture Notes in Computer Science*. Springer, Heidelberg, August 2001.

[WBG10] Christian Wenzel-Benner and Jens Gräf. XBX: external benchmarking extension for the SUPERCOP crypto benchmarking framework. In Mangard and Standaert [MS10], pages 294–305.

[WGZ+16] Lei Wang, Jian Guo, Guoyan Zhang, Jingyuan Zhao, and Dawu Gu. How to build fully secure tweakable blockciphers from classical blockciphers. In Cheon and Takagi [CT16], pages 455–483.

[WHN+12] Hongjun Wu, Tao Huang, Phuong Ha Nguyen, Huaxiong Wang, and San Ling. Differential attacks against stream cipher ZUC. In Wang and Sako [WS12], pages 262–277.

[WIK+08] Dai Watanabe, Kota Ideguchi, Jun Kitahar, Kenichiro Muto, Hiroki Furuichi, and Toshinobu Kaneko. Enocoro-80: A hardware oriented stream cipher. In *The Third International Conference on Availability, Reliability and Security — ARES 08*, pages 1294–1300, 2008.

[WN95] David J. Wheeler and Roger M. Needham. TEA, a tiny encryption algorithm. In Preneel [Pre95], pages 363–366.

[WS12] Xiaoyun Wang and Kazue Sako, editors. *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*. Springer, Heidelberg, December 2012.

[WSD+99] David Wagner, Leone Simpson, Ed Dawson, John Kelsey, William Millan, and Bruce Schneier. Cryptanalysis of ORYX. In Stafford E. Tavares and Henk Meijer, editors, *SAC 1998: 5th Annual International Workshop on Selected Areas in Cryptography*, volume 1556 of *Lecture Notes in Computer Science*, pages 296–305. Springer, Heidelberg, August 1999.

[WSK97] David Wagner, Bruce Schneier, and John Kelsey. Cryptanalysis of the cellular encryption algorithm. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 526–537. Springer, Heidelberg, August 1997.

[Wu16] Hongjun Wu. ACORN: A lighweight authenticated cipher (v3). Candidate for the CAESAR Competition. See also https://competitions.cr.yp.to/round3/acornv3.pdf, 2016.

[WW05] Ralf-Philipp Weinmann and Kai Wirt. Analysis of the DVB Common Scrambling Algorithm. In David Chadwick and Bart Preneel, editors, *Communications and Multimedia Security: 8th IFIP TC-6 TC-11 Conference on Communications and Multimedia Security, Sept. 15–18, 2004, Windermere, The Lake District, United Kingdom*, volume 175 of *IFIP – The International Federation for Information Processing*, Boston, MA, 2005. Springer US.

[WWBC14] Long Wen, Meiqin Wang, Andrey Bogdanov, and Huaifeng Chen. Multidimensional zero-correlation attacks on lightweight block cipher hight: Improved cryptanalysis of an iso standard. *Information Processing Letters*, 114(6):322 – 330, 2014.

[WWJ16] Ning Wang, Xiaoyun Wang, and Keting Jia. Improved impossible differential attack on reduced-round LBlock. In Soonhak Kwon and Aaram Yun, editors, *ICISC 15: 18th International Conference on Information Security and Cryptology*, volume 9558 of *Lecture Notes in Computer Science*, pages 136–152. Springer, Heidelberg, November 2016.

[WZ11] Wenling Wu and Lei Zhang. LBlock: A lightweight block cipher. In Lopez and Tsudik [LT11], pages 327–344.

[YKPH11] Huihui Yap, Khoongming Khoo, Axel Poschmann, and Matt Henricksen. EPCBC - a block cipher suitable for electronic product code encryption. In Dongdai Lin, Gene Tsudik, and Xiaoyun Wang, editors, *CANS 11: 10th International Conference on Cryptology and Network Security*, volume 7092 of *Lecture Notes in Computer Science*, pages 76–97. Springer, Heidelberg, December 2011.

[Yuv97] Gideon Yuval. Reinventing the Travois: Encryption/MAC in 30 ROM bytes. In Biham [Bih97], pages 205–209.

[YZS+15] Gangqiang Yang, Bo Zhu, Valentin Suder, Mark D. Aagaard, and Guang Gong. The simeck family of lightweight block ciphers. In Güneysu and Handschuh [GH15], pages 307–329.

[ZBL+15] Wentao Zhang, Zhenzhen Bao, Dongdai Lin, Vincent Rijmen, Bohan Yang, and Ingrid Verbauwhede. RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms. *Science China Information Sciences*, 58(12):1–15, 2015.

[ZSX+14] Bin Zhang, Zhenqing Shi, Chao Xu, Yuan Yao, and Zhenqi Li. Sablier v1. Candidate for the CAESAR Competition. See also https://competitions.cr.yp.to/round1/sablierv1.pdf, 2014.

[ZWW+14] Lei Zhang, Wenling Wu, Yanfeng Wang, Shengbao Wu, and Jian Zhang. LAC: A lightweight authenticated encryption cipher. Candidate for the CAESAR Competition, 2014.

[ZXF13] Bin Zhang, Chao Xu, and Dengguo Feng. Real time cryptanalysis of bluetooth encryption with condition masking (extended abstract). In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 165–182. Springer, Heidelberg, August 2013.

A List of Ligthweight Ciphers from the Industry

A.1 Industry-Designed Stream Ciphers

A5/1. The exact design date of this algorithm is unclear but a first approximation of its inner workings was published in 1994 [And94]. It generates a keystream from a 22-bit IV along with a 64-bit key using three different LFSRs whose lengths add up to 64 bits. Practical attacks have been implemented using time-memory trade-offs exploiting the fact that the update function of the internal state is not bijective [Gol97, BSW01]. The most time efficient of those needs only 2^{24} simple steps provided that a significant (but practical) pre-computation was performed. Furthermore, 10 bits of the key were always set to 0 in many implementations. The 2G GSM protocol still uses this algorithm.

A5/2. A cipher somewhat similar similar to A5/1 but even weaker was intended to be used in countries targeted by American export restrictions. It is called A5/2. It is vulnerable to ciphertexts only attacks with complexity 2^{16} using redundancy introduced by error correcting codes. It requires a one-time pre-computation of practical complexity. Unfortunately, interoperability imposed the implementation of this algorithm on devices supposed to run A5/1 instead, thus making downgrade attacks possible [BBK03, BBK08].

A5-GMR-1 and A5-GMR-2. Satellite phones have their own protocols and, therefore, use their own cryptographic algorithms. The two algorithms used, A5-GMR-1 and A5-GMR-2, were reverse-engineered by Driessen *et al.* in [DHW+12]. Those are very different from one another but both are easily attacked.

- A5-GMR-1 is a variant of A5/2 with an internal state consisting in 4 LFSRs with a total size of 82 bits. Those are clocked irregularly, much like in A5/2. It can be attacked using only known ciphertexts by inverting 2^{21} triangular matrices of size 532×532, which requires roughly $2^{21} \times 532^2/2 \approx 2^{38.1}$ simple operations. A significant but practical pre-computation step is necessary.

- A5-GMR-2 is a byte oriented stream cipher with a much more sophisticated structure based on 3 different components denoted $\mathcal{F}, \mathcal{G}$ and $\mathcal{H}$ by Driessen *et al.*. Surprisingly, $\mathcal{H}$ uses the S-Box S_2 and S_6 of the DES. A practical attack with very low data and time complexity is presented in [LLLS14]. It requires guessing at most 32 bits using only 1 frame of 15 bytes for an average complexity of 2^{28}.

Atmel Ciphers. The stream ciphers used by the SecureMemory, CryptoMemory and CryptoRF families of products from Atmel are similar to one another. They are proprietary algorithms which were reverse-engineered and attacked by Garcia *et al.* in [GvRVWS10]. Other more powerful attacks were later proposed by Biryukov *et al.* [BKZ11] breaking the cipher of SecureMemory in time $2^{29.8}$ using 1 frame and the cipher of CryptoMemory in time 2^{50} using 30 frames and about 530 Mb of memory. The ciphers rely on 3 NLFSRs with a total size of a bit more than 100 bits. The attacks found by both Garcia *et al.* and Biryukov *et al.* were successfully implemented.

Crypto-1. It is a stream cipher used by the *Mifare classic* line of smartcards of NXP. It was reverse-engineered by Nohl *et al.* in [NESP08] and was subsequently attacked by many teams [CNO08, Gol13] with a time complexity as low as 2^{32}. It has been used at least since 1998 but the exact date of its design is unclear. It is based on a 48-bit LFSR combined with several non-linear Boolean functions.

Content Scrambling System (Css). In order to implement Digital Rights Managements (DRMs), the content of DVD discs is encrypted. This encryption used to be performed with a stream cipher called Css. It uses two 17- and 25-bit long LFSRs to generate two 8-bit words in parallel. These are afterwards added modulo 2^8 to obtain a byte of keystream. However, unlike in most stream ciphers, this key stream is not simply XORed with the plaintext. Instead, the plaintext first goes through an 8-bit bijective S-Box whose result is added to the keystream to obtain the ciphertext. This operation is sometimes called the *mangling step*. A full description is available in [BD04] and in [PMA07]. Several powerful attacks target the protocol using this stream cipher. However, given its key length of 40 bits, the cipher alone is vulnerable to a brute-force search of time complexity 2^{40}.

Common Scrambling Algorithm (Csa-SC). The Common Scrambling Algorithm is used to secure digital television broadcast. It cascades two ciphers, as described in [WW05]. The first is a block cipher which we call Csa-BC and which is described below. The second is a stream cipher which we call Csa-SC. The stream cipher is based on two FSRs consisting of twenty 4-bit cells each and a combiner with memory. The feedback function of the registers involves, among other things, several 5×2 S-Boxes. The combiner uses addition modulo 2^4 to extract 2 bits of keystream from its internal state and the two shift registers at each clock cycle. In [WW05], several undesirable properties are presented. For example, the keystream often has very short cycles. It is also possible to recover the secret key by solving about 2^{28} systems of 60 linear equations with 40 unknowns which must take at most $2^28 \times 60^3 \approx 2^{45.7}$.

Dsc. The DECT[11] Standard Cipher, usually abbreviated into Dsc, is a stream cipher used to encrypt the communications of cordless phones. First, attacks targeting the protocol using it and its flawed implementation were presented in [LST+09]. It was subsequently reverse-engineered and its attackers found practical attacks requiring only about 2^{15} samples of keystream and 2^{34} trial encryptions which take a couple of hours on a standard computer to recover the key [NTW10]. It is described by the authors of this paper as being "an asynchronous stream cipher with low gate complexity that takes a 64-bit secret key and a 35-bit initialization vector." Its structure, based on irregularly clocked LFSRs, is reminiscent of that of A5/1.

E0. The privacy of the Bluetooth protocol is now based on the AES but it used to rely on a custom stream cipher called E0. Its 128-bit internal state is divided into 4 LFSRs

[11] DECT stands for "Digital Enhanced Cordless Telecommunications".

and its filter function has its own 2-bit memory. A description of E0 can be found in the papers presenting attacks against it such as [FL01, LV04, LMV05]. Lu *et al.* found an attack which recovers the secret key using the first 24 bits of $2^{23.8}$ frames and with 2^{38} computations.

Hitag2 ; Megamos. These stream ciphers are used in the *car immobilizers* implemented by different car manufacturers. These devices prevent a car engine from starting unless a specific transponder is close to them. While initially kept secret, the first was published by Wiener[12] and the second was reverse-engineered by Verdult *et al.* [VGE13]. They are both stream ciphers with a small internal state of 48 and 57 bits respectively. These small sizes and other weaknesses in the ciphers themselves and in the protocols using them lead to practical attacks against the devices relying on these algorithms for security. For example, it is possible to attack a car key using Hitag2 using 1 min of communication between the key and the car and about 2^{35} encryptions. The secret key of Megamos can be recovered in time 2^{48} but more powerful attacks are possible when we take into account the key management method of the devices using it.

iClass. Formally, iClass is family of smartcards introduced in 2002. The stream cipher it uses was reverse-engineered and attacked by Garcia *et al.* in [GdKGV14]. It has a 40-bit internal state. The cryptanalysts who reverse-engineered it presented attacks against this cipher in the same paper. By recording 2^{22} authentication attempts, the key can be recovered using 2^{40} trial encryptions.

Kindle Cipher (PC1). This stream cipher was first published on Usenet by Alexander Pukall in 1997, meaning that this algorithm was not technically designed in the industry. However, it was not designed by academics and Amazon used it at least up until 2012 for the DRM scheme protecting its e-book using the MOBI file format. It uses a 128-bit key and a separate 24-bit internal state updated using different operations, including modular multiplications. The keystream is generated byte by byte. It has been broken by Biryukov *et al.* [BLR13] using e.g. 2^{20} known plaintexts and a time of 2^{31}. Even practical known-ciphertexts attacks are possible in some contexts.

Oryx. While A5/1 secures GSM communications in Europe, the stream cipher ORYX was chosen by the Telecom Industry Association Standard (TIA) to secure phone communications in north America. A description of the algorithm can be found in [WSD+99] where practical attacks are presented. It uses a 96-bit key, a 96-bit internal state consisting of three 32-bit LFSRs, and an 8-bit S-Box which changes every time. It is possible to attack it in time 2^{16} using 25 bytes of known plaintext.

A.2 Industry-Designed Block Ciphers

CMEA This block cipher was used by the TIA to secure the transmission of phone numbers across telephone lines. A good description of this algorithm is provided in [WSK97] which, incidentally, describes an attack against the full cipher. It encrypts a block of an arbitrary number of bytes — although in practice those were usually 2 to 6 bytes long — using a 64-bit key. It is vulnerable to a known plaintext attacks requiring only 40–80 blocks of data and taking a time between 2^{24} and 2^{32} encryptions. Its 8-bit S-Box seems to contain a hidden structure [Per17].

[12] While this first publication is mentioned for example by Verdult *et al.* in [VGB12], we were not able to find a copy of it. Nevertheless, the specification of Hitag2 can be found in [VGB12].

Cryptomeria. It is a block cipher nicknamed "C2" in the literature. It shares the same structure as the DES: it encrypts 64-bit blocks using a 56-bit key and uses a 32-bit Feistel function. It works by mixing in a 32-bit subkey with a modular addition, then use one 8-bit S-Box call followed by a 32-bit linear permutation. The S-Box is secret, so an S-Box recovery attack has been proposed [BKLM09]. The same paper presents a key recovery with time complexity 2^{48}. This algorithm was intended from the start to be used by "things", namely DVD players (in which case it can be seen as a successor of CSS) and some SD cards. In total, 10 rounds are used; which means that only 10 S-Box calls are needed to encrypt one 64-bit block compared to, say, the 160 calls needed to encrypt one 128-bit block using AES-128.

Common Scrambling Algorithm (Csa-BC). The Common Scrambling Algorithm uses a stream cipher (described above) and a block cipher which we call CSA-BC. It encrypts a 64-bit block using a 64-bit key. Its structure is reminiscent of a generalised Feistel network using eight 8-bit branches. The Feistel functions are based on a unique random-looking 8-bit S-Box B and a variant defined as $\sigma \circ B$, where σ is a simple bit permutation. An encryption consists in 56 rounds. A full specification is given in [WW05]. To the best of our knowledge, there is no attack other than brute-force against this cipher.

Dst40. This algorithm was reverse-engineered from partial information disclosed in a patent and from a physical device implementing this block cipher [BGS+05]. It was used by RFID transponders sold by Texas Instrument. They were used in car immobilizers and for electronic payment. The cipher itself encrypts a 40-bit block with a 40-bit key using 200 rounds of an unbalanced Feistel network. The Feistel function maps 38 bits of internal state and a 40-bit subkey to a 2-bit output by nesting several Boolean functions. Due to its key size of 40 bits, a brute-force search is practical.

Keeloq. It is a so-called "code-hopping encoder". A US patent was filed in 1993 and eventually granted in 1996 [BSK96] but it was designed earlier, around 1985 [Lea14]. Using modern terminology, it is a 32-bit block cipher which uses a 64-bit key. It was first kept secret but its specification was leaked in 2006. Using this information, several teams presented practical attacks against devices using this algorithm [IKD+08]. For example, the key be recovered using 2^{16} known plaintexts and $2^{44.5}$ encryptions. Far more powerful side-channel attacks have also been proposed against commercial implementations of the cipher [EKM+08]. This ciphers was still in use when these attacks were found, 20 years after its design.

A.3 Industry-Designed Macs

SecurID MAC. A SecurID is small hardware token used for authentication and designed by SDTI (which was later bought by RSA Security). It displays a 6 digit number which changes every minute. It is based on a 64-bit MAC described for example in [BLP04]. This paper also presents attacks against the algorithm. The details of the MAC were initially kept secret but were eventually leaked which lead to the attacks of Biryukov *et al.*. These were later sped up by Contini *et al.* [CY04] to obtain a time complexity of about 2^{44} MAC computations.